UNISA college of science, engineering and technology

A FRAMEWORK FOR ACTIVITY-BASED COMPUTER PROGRAMMING INSTRUCTION IN A BLENDED LEARNING ENVIRONMENT

by

EMMANUEL FREEMAN
(Student Number: 60870540)

Submitted in accordance with the requirements for
the degree of Doctor of Philosophy

in the subject

COMPUTER SCIENCE

at the

University of South Africa

Supervisor: Prof. Ian D. Sanders

Co-supervisor: Prof. Bester Chimbo

March 2021

DEDICATION

I wish to dedicate this thesis to the God Almighty and to my parents Nana David Akewsi Freeman and Madam Rose Ama Aboagyewaa for the supernatural and physical provisioning of my educational ladder.

ACKNOWLEDGEMENTS

My sincere appreciation is to the God Almighty for His marvelous grace that has brought me this far.

Also, my profound gratitude goes to my lovely wife, Stephanie Nyamekye Freeman and son Jason Akorabeng Freeman for the encouragement, support and ardent sacrifices they afforded me throughout my academic journey.

I would also like to express my deepest gratitude to my supervisors Prof. Ian D. Sanders and Prof. Bester Chimbo for their immeasurable directions, motivation, and support rendered to me throughout my four years' journey with the University of South Africa. God richly bless you Prof. Ian and Prof. Bester.

I am also appreciative of the provision of financial support by the Bursary Council of UNISA throughout my study with UNISA. To my current employer, Ghana Communication Technology University (GCTU), I also appreciate the support and privilege granted to me to pursue my PhD program.

To Prof. Kirstin Krauss, Prof. Alfred Coleman and Estelle De Kock all of UNISA, and my subordinates at the Centre for Online Learning and Teaching (COLT) at GCTU, I am very grateful for your support and words of encouragement.

ABSTRACT

This study develops a framework for an activity-based computer programming instructional approach in a blended learning environment of higher education institutions. The driving force for this study is the poor performance of most undergraduate students in the domain of computing science programs to comprehend and write computer programs to solve real-world problems. In addition, a fundamental problem with most HEIs in undergraduate computer science educational programmes is how to teach programming and how students can comprehend and write computer programs. Hence, the study developed a framework and a curriculum mediation for computer programming instruction in Higher Education Institutions (HEIs) in Ghana.

The action research approach was used to conduct the research which spanned through three different semesters from September 2019 to October 2020. The mixed method approach was used to capture data from 300 students and nine lecturers in three different HEIs in Ghana and different computer programming classes within a blended learning environment.

Findings from the study provided a curriculum mediation and a framework indicating the various components of activity-based computer programming instructional approach within a blended learning environment that enhances learning gains for students. It was found that instructional activities, cognitive activities, skills development, assessment and feedback, curriculum and technology mediation, and learner support activities have positive significant effect on students' learning gains especially with the comprehension of computer programming syntax and semantics. Moreover, the study found that students' learning journey of computer programming comprehension needs continuous learner support and engagements using activity-based instructional approach within a blended environment.

Based on the findings, recommendations to HEI's managers, administrators, the National Accreditation Board, the Ministry of Education and the National Council for Tertiary Education were made to make reforms in computer programming curricula through the use of an activity-based pedagogic approach to advance students' programming skills, knowledge and employability opportunities.

DECLARATION

I declare that this study, **A FRAMEWORK FOR ACTIVITY-BASED COMPUTER PROGRAMMING INSTRUCTION IN A BLENDED LEARNING ENVIRONMENT**, is my own work and that all the sources that I have used or quoted have been indicated and acknowledged by means of complete references. I further declare that I have not previously submitted this work, or part of it, for examination at Unisa for another qualification or at any other higher education institution.

________________________ 30th March, 2021.

Signature Date

Table of Contents

DEDICATION ii
ACKNOWLEDGEMENTS iii
ABSTRACT iv
DECLARATION v
List of Tables xiii
List of Figures xv
List of Abbreviations xvii
Terms and Definitions xix
CHAPTER 1 1
INTRODUCTION AND THESIS OVERVIEW 1
1.1 Introduction and Background 1
1.2 Problem Statement 3
1.3 Research Objectives 5
1.4 Research Questions 5
1.5 Brief Research Methodology 6
1.6 Research Contribution 7
1.7 Thesis Structure 8
CHAPTER 2 10
REVIEW OF LITERATURE 10
2.1 Introduction 10
2.2 Educational Paradigms in HEIs. 10
2.3 Computer Science Education 14
2.4 The Concept of Computer Programming 16
2.5 Teaching and Learning of Computer Programming 19
2.5.1 Problems of Teachings and Learning of Computer Programming 20
2.5.2 Reasons Why Learning Computer Programming is a Problem among HEIs Students 21
2.5.3 Addressing the Problems in Teaching and Learning of Computer Programming among HEIs Students 23
2.6 Instructional Approaches in Teaching Computer Programming 25
2.6.1 Problem-based Approach 25
2.6.2 Puzzle-based Learning Approach 28
2.6.3 Pair Programming Approach 30
2.6.4 Pre-recorded Lectures Approach 31
2.6.5 Game-based Programming 32
2.7 Activity-based Teaching and Learning 34

2.7.1 Activity-based Teaching and Learning: What is it? 34
2.7.2 Philosophical Assumptions of an Activity-based Learning Approach. 35
2.7.3 Preparing to Teach Using an Activity-based Learning Approach. 37
2.7.4 Activity-based Learning in Higher Education 40
2.7.5 The Relevance of Activity-based Learning (ABL) in Higher Education 42
2.8 Teaching and Learning within a Blended Learning Environment 43
2.8.1 Blended Learning: What is it? 43
2.8.2 Theoretical Rationales of Blended Learning. 44
2.8.3 Blended Learning in HEIs. 45
2.8.4 The Realities between the Traditional Method of Teaching Vs. Blended Learning Approach 46
2.9 Review of Existing Models of Teaching and Learning within Blended Learning Environments of HEIs 48
2.9.1 E-moderation Model 49
2.9.2 The Learning Mix Model 53
2.9.3 Learning Ecology Model 55
2.9.4 Blended Learning Continuum Model 56
2.10 Technology Enhanced Learning (TEL) 60
2.10.1 Learning Management System (LMS) 61
2.10.2 Engagement in the Classroom 62
2.10.3 Engagement Beyond the Classroom 63
2.10.4 Assessment and Feedback in TEL 63
2.11 Smart Learning 64
2.12 Conclusion 66
CHAPTER 3 67
THEORETICAL AND CONCEPTUAL FRAMEWORK 67
3.1 Introduction 67
3.2 Learning Theories 67
3.2.1 Constructivism Theory 68
3.2.2 Behaviourism Theory 69
3.2.3 Cognitive Learning Theory 70
3.2.4 Social Constructivism Theory 71
3.3 Theoretical Framework of the Study 71
3.3.1 The Cognitive Presence 74
3.3.2 The Social Presence 74
3.3.3 Teaching Presence 75
3.4 Conceptual Framework of the Study 76

3.4.1 Development of the Conceptual Framework 76
3.5 The Conceptual Framework 79
3.5.1 Community of Inquiry Component 79
3.5.2 Activity-based Approach Component 81
3.5.3 Learning Gains 81
3.5.4 Curriculum Mediations 82
3.5.5 Feedback and Assessment 83
3.5.6 Students' Skills Development 83
3.6 Conclusion 84
CHAPTER 4 85
RESEARCH METHODOLOGY 85
4.1 Overview of the Methodological Approach 85
4.2 Philosophical Perspective of the Research 85
4.3 The Justification for Adopting the Constructivist Theory 87
4.4 Action Research (AR) Design 89
4.4.1 Action Research Design Processes Used 90
4.4.2 Action Research Timelines 94
4.5 Research Hypotheses 96
4.6 Application of the Mixed Method Approach in the Action Research 96
4.7 Research Approach 98
4.8 Research Design and Methods 99
4.9 Framework of Research Design and Methods for the Study 99
4.10 Research Population and Sampling 105
4.10.1 Sampling Procedure 105
4.10.2 Population 105
4.10.3 Sample Frame 107
4.10.4 Sample Size 108
4.10.5 Sample Technique 109
4.11 Data Collection 110
4.11.1 Questionnaire Administration 112
4.11.2 Interviews Administration 114
4.11.3 Focus Group Discussions 115
4.12 Data Analysis 120
4.12.1 Structural Equation Modelling (SEM) 120
4.12.2 Multivariate Normality 121
4.12.3 Multicollinearity 122
4.12.4 Sample Size 122

4.12.5 Positive Definiteness 122
4.12.6 Univariate Normality 122
4.13 Reliability and Validity of the Research Data 123
4.14 Ethical Considerations 125
4.15 Summary of Research Methodology Chapter 126
CHAPTER 5 128
PRESENTATION OF RESULTS AND ANALYSIS 128
5.1 Introduction 128
5.2 Analysis of First Semester Results of the Action Research 128
5.2.1 Response Rate 128
5.2.2 Reliability Analysis (Data for Semester 1) 129
5.2.3 Demographics of Respondents 130
5.2.4 Computer Programming Courses Enrolled by the Respondents 131
5.2.5 Characteristics of Respondents' Computing Literacy 132
5.2.6 Analysis of the Current Activity-based Teaching and Learning Approaches Used in Computer Programming Instruction (First Semester) 133
5.2.7 One-Way ANOVA Test of the Activity-based Approaches Used 135
5.2.8 Challenges of Learning Programming 136
5.2.9 One-Way ANOVA Test on the Challenges of Learning Programming 137
5.2.10 Findings from the Qualitative Analysis of the Study for the First Semester 137
5.2.10.1 Profile of the Participants 138
5.2.10.2 The Students' Learning Experience and Perception on the Current Activity-Based Approach Used in Computer Programming Instruction 138
5.2.10.3 Relevance of the First Semester Findings on the ABLA Framework Development 142
5.3 Analysis of Second Semester Activities of the Action Research 143
5.3.1 Design of Curriculum Mediation Based on Pedagogic Approaches that Supports Activity-Based Learning for Instructing Computer Programming in HEIs. 143
5.3.2 Developed Curriculum for Computer Programming 144
5.3.3 Aims and Summary of the Course 145
5.3.4 Intended Module / Course Learning Outcomes 145
5.3.5 Indicative Contents 145
5.3.6 Methods of Assessment 146
5.3.7 Suggested Activity-Based Pedagogic Approaches 148
5.4 Development of the Activity-Based Programming Instructional Platform (ABPI) 151
5.4.1 User Interface Design of ABPI System 153
5.4.2 The Activities Interface 155
5.4.3 The Courses and Topics Interface 156

5.4.4 Enrolled Instructors and Courses....157
5.4.5 Learning Management System (LMS) Used....158
5.5 Analysis of the Curriculum Mediations Developed....160
5.6 Reflections on the Curriculum Mediation and ABPI for Computer Programming Instruction.....162
5.7 Relevance of the Second Semester Findings on the ABLA Framework Development163
5.8 Conclusion....163
CHAPTER 6 165
DEVELOPMENT OF THE FRAMEWORK FOR ACTIVITY-BASED COMPUTER PROGRAMMING INSTRUCTION....165
6.1 Introduction....165
6.2 Results of Third Semester Activities of the Action Research....165
6.2.1 Characteristics of the Contructs....166
6.2.2 .One-Way ANOVA Test to Determine Differences in Mean Scores by the HEIs Based on the Constructs....167
6.3 Designing the ABLA Framework....168
6.3.1 Implementation of the Conceptual Design of theABLA Framework....171
6.4 Quantitative Analysis of the Results of the ABLA Framework....172
6.4.1 Reliability of the Instrument....172
6.4.2 Exploratory Factor Analysis....173
6.4.3 Validity Test Using Exploratory Factor Analysis....174
6.4.4 Principal Component Analysis: Communalities....174
6.4.5 Total Variance Explained....179
6.4.6 Analysis of the Structural Equation Model....181
6.4.7 Confirmatory Factor Analysis (CFA)....182
6.4.8 Factor Loadings and Critical Ratios....182
6.4.9 Structural Model for Confirmatory Factor Analysis....183
6.4.10 Common Method Bias....184
6.4.11 Model Fit for Confirmatory Factor Analysis....185
6.4.12 Cut-off Criteria....186
6.4.13 Confirmatory Factor Analysis Measures....187
6.4.14 Validity Analysis....188
6.4.15 Assessing the Effects....189
6.4.16 Structural Model of the Combined Effect....191
6.4.16.1 Model Fit Indices for Combined Effect....192
6.5 Qualitative Analysis of the Study During the Third Semester....194
6.5.1 Profile of the Participants (Students)....194
6.5.2 Interview Responses (Students)....195

6.5.3 The Students' Learning Experience and Perception of the ABLA Framework Used in Computer Programming Instruction **196**

6.5.4 Qualitative Analysis of Instructors on the ABLA Framework for Instructing Computer Programming in HEIs. **198**

6.5.4.1 Demographics of Instructors **199**

6.5.5 Thematic Analysis of Instructors' Experiences on ABLA Framework for Computer Programming Instruction. **200**

6.5.5.1 Activity-Based Teaching Experience within Blended Learning Environment **201**

6.5.5.2 Computer Programming Teaching and Learning Experience **202**

6.5.5.3 Satisfaction **203**

6.6 Reflections and Re-designing of the ABLA Framework for Computer Programming Instruction **203**

6.7 Reflections on the Students' Experiences of the Blended Learning (BL) Environments and Provisioning of Learner Support **205**

6.8 Relevance of the third Semester Findings on the ABLA Framework Development **207**

6.8 General Reflections of Data Analysis and Presentation of Results for First, Second, and Third Semesters of the AR Research **208**

6.9 Testing and Evaluation of ABLA Framework Per the Hypotheses **209**

6.10 Conclusion Derived from the Evaluated Hypotheses **211**

6.10 A New Framework for Activity-Based Programming Instruction within a Blended Learning Environment **213**

6.11 Conclusion **215**

CHAPTER 7 **217**

DISCUSSIONS, CONCLUSIONS AND RECOMMENDATIONS **217**

7.1 Introduction **217**

7.2 Summary of Research Findings **218**

7.2.1 Research Objective 1 **218**

7.2.1.1 Summary of Research Findings **218**

7.2.1.2 Theoretical Implications of the Research **219**

7.2.1.3 Practical Implications **220**

7.2.1.4 Policy Implications **223**

7.2.2 Research Objective 2 **224**

7.2.2.1 Summary of Research Findings **224**

7.2.2.2 Practical Implications on the Findings **225**

7.2.2.3 Policy Implications of the Findings **225**

7.2.3 Research Objective 3 **226**

7.2.3.1 Summary of Research Findings **226**

7.2.3.2 Practical Implications on the Findings **228**

7.2.3.3 Theoretical Implications **229**

7.2.3.4 Policy Implications **230**
7.2.4 Research Objective 4 .... **230**
7.2.4.1 Summary of Research Findings **230**
7.3 Literature Gaps Addressed **231**
7.4 Summary of Research Objectives, Methodology, Findings, Implications and Research Gaps Addressed **234**
7.5 Recommendations **239**
7.6 Suggestions for Future Research **240**
7.7 Conclusion **240**
References **242**
Appendix A: Ethical Clearance **277**
Appendix B: Questionnaire for Respondents (Students) – First Semester **279**
Appendix C: Students' Interview Guide (First Semester) **283**
Appendix D: Faculty Interview Guide on the Curriculum Design (Semester 2) **284**
Appendix E: Questionnaire for Respondents (Students) – After Second Semester **285**
Appendix F: Students Interview Guide on the ABLA Framework (Semester 3) **291**
Appendix G: Faculty Interview Guide on the ABLA Framework (Semester 3) **292**
Appendix H: Permission Letters **295**
Appendix I: Communalities of the SEM of the Data **298**
Appendix J: Assessment of Normality **299**
Appendix K: Sample Codes for the ABPI Platform **302**
Appendix L: Poster Presentation at SAICSIT Conference (2019), South Africa **321**

List of Tables

Table 2.1 Framework Distinguishing Face-to-Face Delivery and Virtual Delivery (Chew, Jones and Turner, 2008)....48

Table 2.2 Similarities of Allen et al. (2007) and Jones (2006) Continuum Model of Blended Learning. 58

Table 4.1 Action Research Timelines....94

Table 4.2: Framework of Research Methodology Used for the Study....101

Table 4.3 Population of the Study....106

Table 4.4: Sampling Size....109

Table 4.5: Research Questions and Instruments Adopted....111

Table 4.6: Demonstration of Analytical Methods and Data Analysis Used to Address the Research Questions....116

Table 5.1: Response Rate of the Sample....129

Table 5.2: Reliability Analysis....129

Table 5.3 Characteristics of Respondents....130

Table 5.4 Computer Programming Demographics of Respondents....132

Table 5.5 Characteristics of Respondents' Computing Literacy....133

Table 5.6: Activity-Based Approaches Employed....134

Table 5.7 One-Way Anova Test of The Activity-Based Approaches Used....135

Table 5.8 Challenges of Activity-Based Approach in Programming Instruction....136

Table 5.9: One-Way Anova Test on the Challenges of Activity-Based Approach in Computer Programming Instructions....137

Table 5.10: Demographics of Interviewees....138

Table 5.11: Course Codes and Titles....144

Table 5.12 Assessment Methods....146

Table 5.13 Suggested Activity-Based Pedagogic Approaches....148

Table 6.1 Characteristics of the Variables....166

Table 6.2 One-Way Anova Test to Determine Differences in Mean Score by the HEIs 168

Table 6.3: Reliability Analysis 173

Table 6.4: KMO And Bartlett's Test 174

Table 6.5: Pattern Matrix[a] 175

Table 6.6: Total Variance Explained 180

Table 6.7: Common Method Bias 184

Table 6.8 Model Fit Information For CFA 185

Table 6.9: Cut-off Criteria 186

Table 6.10 CFA Measures 187

Table 6.11 Convergent and Discriminant Validity 188

Table 6.12: Assessing the Effects 191

Table 6.13: Model Fit Indices for Combined Effect 193

Table 6.14: Path Analysis for Combined Effect 193

Table 6.15: Demographics of Interviewees (Students) 195

Table 6.16: Demographics of the of Instructors 199

Table 6.17: Thematic Analysis of Instructors' Experiences on the ABLA Framework for Programming Instruction 200

Table 6.18 Conclusion Derived from the Research Hypothesis 211

Table 7.1: Summary of Research Objectives, Methodology, Findings, Implications and Research Gaps Addressed 234

List of Figures

Figure 2.1: The Demands on Higher Education Amended from (Turban, 2002)............12

Figure 2.2 E-learning System Architecture (Protus) Derived from Milićević et al. (2017) 14

Figure 2.3 SDLC, Summary by Techbeamers.com, (2019) 18

Figure 2.4 Example of Problem-based Learning in Programming……………………...27

Figure 2.5 A Puzzle Problem For A* Search Algorithm...29

Figure 2.6 Pair Programming Concept (Source: Anarsolutions, 2020)...........................30

Figure 2.7 Box Island Game Tool For Programming...34

Figure 2.8: Abraham Maslow's Hierarchical Theory of Need and Pedagogic Effect (Source: Alexander, 2006 And Chew, Jones and Turner, 2008)...................................50

Figure 2.9: Salmon's Model of E-Moderation, Salmon (2000, 2001) 51

Figure 2.10: The Blended Learning Mix Model (Kaur And Ahmed, 2006)....................53

Figure 2.11: Learning Modalities (Wenger And Ferguson, 2006).................................55

Figure 2.12: Continuum of Blended Learning (Jones, 2006) 57

Figure 2.13 Technology Enhanced Learning Model Within a Blended Learning Environment, Author (2019)...60

Figure 2.14 Approaching Smart Learning Environments (Bosch Software Inc., 2019) ..65

Figure 3.1: The Community of Inquiry Model, (Garrison, Anderson And Archer, 2010 P.2)...72

Figure 3.2: Schematic Processes for Designing the Conceptual Framework, Author (2019)..78

Figure 3.3: Conceptual Framework of the Study, Author (2019) 79

Figure 4.1 Different Worldviews of Research And Methods Used, Source: Creswell, (2014)...88

Figure 4.2: Action Research Model, Source: De Villiers (2005b).................................90

Figure 4.3: Action Research Design Process Model, Author (2019) 91

Figure 5.1 Progress View Towards ABLA Framework Development as at Semester One .... **ERROR! BOOKMARK NOT DEFINED.**

Figure 5.2: Flow Chart of the ABPI System 152

Figure 5.3: Use Case Diagram Of The ABPI System 153

Figure 5.4: Dashboard of the ABPI System (Administrator Interface) 155

Figure 5.5: Sample Activities 156

Figure 5.6: Sample Topics and Courses 157

Figure 5.7: Enrolled Instructors and Assigned Courses 158

Figure 5.8: Sample LMS Platform Used 160

Figure 5.9 Progress View Towards ABLA Framework Development as at Semester 2 163

Figure 6.1 Conceptual Design of the ABLA Framework 170

Figure 6.2 ABLA Framework with the Hypotheses of the Study 181

Figure 6.3 Structural Model for Confirmatory Factor Analysis 183

Figure 6.4: Main Structural Equation Model 190

Figure 6.5: Structural Model of Combined Effect 192

Figure 6.6 Re-design of the ABLA Framework..205

Figure 6.7 Progress View Towards ABLA Framework Development as at Semester 3….. 208

Figure 6.8: Integrated Framework For Activity-Based Programming Instruction......... 214

Figure 6.9 Progress View Towards ABLA Framework Development After Semester 3 215

List of Abbreviations

AB - Activity-based

A&FB - Assessment and Feedback

ABLA - Activity-based Learning Approach

ABLABLE - Activity-based Learning Approach in a Blended Learning Environment

ABPI - Activity-based Programming Instruction

AR - Action Research

AU- Activity-based Approach Used

AVE - Average Variance Extracted

BL - Blended Learning

CA - Cognitive Activities

CFA - Confirmatory Factor Analysis

CFI - Comparative Fit Index

CMS - Course Management System

COI - Community of Inquiry

COLT - Centre for Online Learning and Teaching

COVID - Corona Virus Disease

CR - Construct Reliability

CS - Computer Science

E-learning - Electronic learning

E-mail - Electronic Mail

EFA - Exploratory Factor Analysis

GBP - Game-based Programming

GCTU - Ghana Communication Technology University

HEI - Higher Education Institution

HEIs - Higher Education Institutions

HOD - Head of Department

IA - Instructional Activities

ICT - Information Communication Technology

IFI - Incremental Fit Index

IoT - Internet of Things

ISTE - International Society for Technology in Education

IT - Information Technology

LG - Learning Gains

LMS - Learning Management System
M-learning - Mobile Learning
MMLS - Multimedia Learning Resources
MSV - Maximum Shared Square Variance
MVC - Model View Controller
NAB - National Accreditation Board
NFI - Normed Fit Index
NNFI - Non-Normed Fit Index
ODL - Open Distance Learning
OOP - Object Oriented Programming
OUM - Open University of Malaysia
PBL - Problem-based Learning
PCA - Principal Component Analysis
PLA - Pre-recorded Lectures Approach
PZBL - Puzzle-based Learning
QAA - Quality Assurance Agency
RAD - Rapid Application Development Approach
RMR - Root Mean Square Residual
RMSEA - Root Mean Square Error of Approximation
SA - Social Activities
SD - Skills Development
SDLC - Software Development Life Cycle
SEM - Structural Equation Model
SPSS - Statistical Package for Social Scientists
STEM - Science, Technology, Engineering and Mathematics
TA - Teaching Assistance
TEL - Technology Enhanced Learning
UNISA - University of South Africa
VLE - Virtual Learning Environment

Terms and Definitions

Activity-based Learning: This is an established learning approach of actively engaging students in a classroom or outside the classroom to participate in the learning process through discussions, group or individual reflection, case studies, practical exercises, problem-based learning, etc. (Anwar, 2019; Singal et al., 2018; Margaryan, Collis and Cooke, 2004). The term activity-based learning is alternatively also referenced as active learning (Hazzan, Lapidot and Ragonis, 2014).

Traditional Learning: This is a type of learning where the learner or the student learns mainly from the instructor or the lecturer. This is also known as the traditional face-to-face learning. In this type of learning, the student can directly ask questions, share ideas with the teacher or peers and get feedback right away (Nouri, 2016).

Brick and Mortar Learning Environment: This is a term that connotes a traditional face-to-face delivery of teaching and learning where most of the learning experiences occur at the learner's physical location, i.e. within a building or a classroom. The learner or the student learns mainly from the instructor or the lecturer in the classroom. This is also known as the traditional face-to-face learning.

Online Learning: This is a type of learning acquired with the aid of the Internet. This enables a seamless access to learning resources anywhere and anytime by the learner outside the physical location of the classroom. This approach is also termed e-learning (Olsson and Mozelius, 2016).

Open and Distance Learning (ODL): This is generally an educational delivery via the use of the Internet where education is conveyed to a student outside the physical location of the student's campus. This learning experience enables the learner to learn seamlessly irrespective of the student's location, time and pace (Olsson and Mozelius, 2016).

E-learning: This is a learning approach conducted via the use of electronic technologies and the Internet. This learning approach aids the student or the learner to access learning resources outside the traditional classroom environment usually with the aid of a Learning Management System (LMS) (Huang, Ma and Zhang, 2008; Fisher, Byrne and Tangney, 2016).

Blended Learning: Blended learning is the learning approach that integrates both electronic learning and the traditional learning environment. This is also known as the hybrid learning approach (Kintu, Zhu and Kagambe, 2017).

M-learning: This is a learning approach where learning can be acquired via the use of personal mobile devices. Learning engagements via mobile learning include social and content interactions, learning resources, Technology Enhanced Learning Tools (TEL), etc. The devices used for m-learning include tablets and smart phones (Goodyear and Retalis, 2010).

Computer Programming: This is a process of developing or writing sets of instructions to execute a certain task or problem by the computer. The sets of instructions are termed as the source code which are written in one or more programming languages (Kasame, Pachoen and Manit, 2016).

Pedagogy: This is the art, method, practice and approach of teaching academic concepts or theoretical thoughts (Mishra and Koehler, 2006).

High Level Language: This is a human readable programming language that enables development of programs in a user-friendly development context to solve a problem. Examples include C++, C, Java, PhP, ASP.net, VB.net C#, etc. (Linn, 2009).

CHAPTER 1

INTRODUCTION AND THESIS OVERVIEW

1.1 Introduction and Background

The method of teaching in Higher Education Institutions (HEIs) has gone through a paradigm shift from the traditional face-to-face teaching approach to a technology-driven approach (Moorthy and Arulsamy, 2014). Mavuso and Jere (2020) asserted that the traditional teaching and learning approaches that previously dominated the education industry are presently undergoing a significant transition in the learning paradigm. In order to enhance teaching and learning, several institutions of higher learning have completely adopted a blended learning strategy (i.e. The use of both face-to-face learning and online learning). Additionally, Mavuso and Jere (2020) affirms that it is still difficult to meet the demands of students, particularly in computer science and information technology courses and especially with their expectations of pursuing computing degrees.

Computer science educational systems globally are currently reforming as a result of the drastic changes in societal advancements, socio-economic developments, and political factors. In view of this, there is the need for enhancing the development of HEIs to promote an effective academic activity that solves global problems in teaching and learning in HEIs. In times past, computer science education supported the traditional pedagogic approach of face-to-face teaching and learning. However, recent advancements in information technology and Internet technologies have opened new avenues for researchers to discover new approaches that can lead to an effective approach of teaching computer science programming courses in HEIs (Voronina et al., 2017).

The blended learning environment has come to stay since most HEIs are advocating for an online mode of teaching and learning to make education accessible to everybody especially in the midst of the deadly Covid-19 pandemic (Dhawan, 2020; Blanco et al., 2020). The blended learning environment focuses on the seamless integration of some components of traditional face-to-face teaching and learning and an online-based learning experience. With the blended learning environment, students can prepare for lectures by interacting with the online learning resources which in effect reduces lecture time and increases active engagement in the classroom (Chung, Subramaniam and Dass, 2020; Nouri, 2016; Voronina et al., 2017; Alkhatib, 2018).

The blended learning method includes the development of an interactive module designed for each class with short videos designed to review the learning materials, pre-recorded video or audios and links to related sources for learning needs before the class period. Students can submit assignments online, contribute to forum discussions, do collaborative learning, chat with the lecturer, etc. online. Blended learning adoption requires the student to do personal brainstorming of some key questions, lecture theory, and hands-on exercises, which are communicated to the student using an interactive presentation tool designed to engage the student in lecture activities (Alkhatib, 2018).

The demand for transformations for most higher education institutions globally has necessitated various reforms in higher education institutions' delivery and pedagogic approaches (Chung, Subramaniam and Dass, 2020; Hazzan, Ragonis and Lipidot 2020). Alluding to the transformational agenda of HEIs including pedagogies that could support teaching and learning, the researcher sought to employ the activity-based approach for teaching computer programming within the domain of computing sciences in a blended learning environment.

The activity-based teaching and learning approach focuses on an approach where students are engaged through practice or actions (Anwar, 2019; Singal et al., 2018; Hazzan, Ragonis and Lipidot, 2020; Margaryan, Collis and Cooke, 2004; Ali, 2005). This is, however, in contrast to the instructional approach where the instructor typically relays information or imparts knowledge to students and requires that the student absorbs everything as presented in the classroom. The activity-based approach is known to be a formidable approach for teaching and learning in HEIs (Celik, 2018). This alludes to the study of Silberman (1996, p.2) which for over two decades now has been of significant relevance in higher education (Cherney, 2011). He asserts that "*above all, students need to 'do it', figure things out by themselves, come up with examples, try out skills, and do assignments that depend on the knowledge they already have or must acquire*". Hence the activity-based approach is seen to be a good approach in the teaching of computer science education. In view of this, the approach aids to engage students with practical examples, hands-on interactivities and engagement using case studies or real world problems (An, 2013; Fisher, Byrne and Tangney, 2016, Kosterelioglu and Yapici, 2016).

The researcher sought to use the blended learning environment for the delivery of an activity-based approach for teaching and learning computer programming in HEIs. The rationale for employing the blended learning approach is to enhance teaching and learning in both traditional face-to-face environments and online engagements between students and instructors.

1.2 Problem Statement

A core problem with most HEIs in undergraduate computer science education is how to teach programming and how students can comprehend and code programs (Ahadi, Vihavainen and Lister, 2016; Overmars, 2004; Ali, 2005; Linn, 2009).
Most HEIs teaching approaches have been criticized strongly by Nouri, (2016); and Voronina et al., (2017). These researchers have shed light on the following key problems in HEIs:

1. Students are still using the traditional lecture halls, lecture materials, and conventional assessment methods.
2. Most students see the traditional classroom environments as unexciting and not engaging enough.
3. The teaching pedagogy by most lecturers is not easily adopted by all students, especially students with slow learning capabilities with different learning styles.
4. Most traditional lecture halls are not suited for teaching practically oriented courses requiring skills development and activity learning that involves computer programming and application related exercises.
5. The advent of technology has driven most students towards the use of information technology and smart technologies for learning. In other words, students prefer text books, assignment submission, forum discussions, videos, etc. in a digital format to the hard copy versions or manual approach.
6. The job market requires HEIs to take responsibility for creating a holistic environment that prepares students for the job market and also builds the students' professional development and perpetual learning abilities (Jain, 2010; Mackay, 2010; Quinn, 2009).

The above problems were not different from those identified in the study of Sarpong, Arthur and Amoako, (2013) who asserted that there is a high rate of failure among students in computer programming courses. They asserted that the failure rate could be attributed to the methods used in teaching computer programming at the undergraduate level. Researchers such as Rahmat et al., (2012), Jenkins (2002), Wiedenbeck, LaBelle and Kain (2004) and Sarpong, Arthur and Amoako (2013) asserted that teaching and learning programming presents academic challenges to a lot of students who find programming concepts very difficult to comprehend. Among the challenges faced by most students in computer programming classes are:

1. Lack of understanding of the programming syntax and logics.
2. Superficial understanding of programming questions.

3. Lack of guidance to help solve programming problems.
4. Lack of learner support to aid students in identifying errors, debugging codes and writing codes.
5. Instructors' attitude and personality in the teaching of programming.

Butler and Morgan, (2007) also asserted that undergraduate Information Technology and Computer Science students face a wide variety of academic challenges especially at the first-year level where the foundational computer science programming courses are being taught. These challenges as discovered by Butler and Morgan are not different from the challenges as indicated by Rahmat et al., (2012), Wiedenbeck, LaBelle and Kain (2004) and Sarpong, Arthur and Amoako (2013). This is because most students progressing from high school education to tertiary level have not had prior knowledge in computing programming, resulting in the abysmal performance by most students at the early stages of computer science programmes. The high failure rate in programming has been the trend in most Ghanaian universities (Sarpong, Arthur and Amoako, 2013).

Also, the challenging situations of computer programming instruction still remains the same depicting a defect in the method of teaching the course. Hence, the need for a further study to bridge the gap in the delivery of the course. The situation is however worse when the traditional method of teaching and learning still remains unchanged as the main adopted approach in teaching in HEIs (Roehl, Reddy and Shannon, 2013). Moreover, there is no clear policy to teach computer programming using an activity-based learning approach within a blended environment to support lifelong learning (Kinshuk et al., 2016; Wilcox and Lionelle, 2018).

Furthermore, numerous researchers and educational advocates such as Nouri (2016); Kintu, Zhu and Kagambe (2017); Wulf (2005); Margaryan, Collis and Cooke (2004); and Fisher, Byrne and Tangney (2016) have underscored reforms in HEIs' teaching and learning philosophies. They have advocated the adoption of an active learning philosophy with the use of technology to enhance teaching and learning. It was discovered, however, that previous studies did not employ an activity-based learning approach in the area of computer science education. So, a research gap exists to integrate activity-based learning and blended learning for programming instruction which this study intends to explore.

1.3 Research Objectives

The main objective of this study is to develop a framework for an Activity-based Learning Approach (ABLA) in the instruction of computer programming in a blended learning environment in higher education.

In order to achieve the above main objective, the following specific research objectives were used as a guide for this research:

1. To ascertain the current activity-based learning approaches in computer programming instruction in HEIs in Ghana within a blended learning environment.
2. To develop a curriculum mediation based on pedagogic approaches that could support activity-based learning for instructing computer programming for HEIs in Ghana.
3. To investigate the learning gains of an activity-based learning approach in a blended learning environment for students and lecturers in HEIs of Ghana.
4. To develop a new framework that enhances an activity-based learning approach in a blended learning environment for teaching computer programming in HEIs.

1.4 Research Questions

The primary research question for this study is: How should activity-based learning be incorporated into a blended learning environment in HEIs of Ghana?

The study sought to answer the following secondary research questions to unravel the solution to the primary research question.

1. What are the current activity-based learning approaches in computer programming instructions within a blended learning environment in HEIs of Ghana?
2. What curriculum mediation based on pedagogic approaches could support activity-based learning for instructing computer programming in HEIs of Ghana?
3. What are the learning and teaching gains of an activity-based learning approach in a blended learning environment for students and lecturers in HEIs of Ghana?
4. What new framework could be used to enhance an activity-based learning approach in a blended learning environment for instructing computer programming?

1.5 Brief Research Methodology

The objective of this study is to develop a framework for the implementation of activity-based teaching and learning in HEIs within a blended learning environment among undergraduate students. Also, the study focuses on the students' and lecturers' teaching and learning experiences in the field of computer science education, specifically in computer programming instruction.

The Action Research (AR) approach was used to conduct the research towards the design and development of an activity-based instructional approach for teaching computer programming within a blended learning environment. The concept of AR has been established as a form of a dynamic study of social change (Baskerville, 1999; Cohen, Manion and Morrison, 2000, and Chen and Zelinsky, 2003). The action research approach used to conduct this research traversed through three different semesters from September 2019 to October 2020 among three HEIs in Ghana.

The researcher employed the mixed method within the context of action research design. That is, employing both qualitative and quantitative research approaches (Creswell and Porth, 2018) to make arguments from experiences, observations, interviews, surveying, questioning, and participations among undergraduate students and lecturers in HEIs in Ghana within the scope of three semesters. Three higher education institutions in Ghana were used as case studies for this research. For purposes of confidentiality, these universities are represented as HEI-1, HEI-2 and HEI-3 which are all located in the capital city of Ghana, Accra. The rationale for choosing these HEIs in Ghana is that these institutions are known to be technology-driven universities in Ghana and are known to employ blended learning in their teaching experiences. Also, each of these institutions award computing degrees in computer science, information technology and computer engineering.

The population for this research constitutes all the programming students among the three selected universities. Also, the population for the lecturers constitute all faculty members teaching programming at the three universities. On the other hand, the sample size for the students' population for the research constitutes two hundred students from each of the three universities in Ghana (i.e., 600 students in total) and nine lecturers respectively (i.e. 3 lecturers from each of the three HEI). Purposive sampling techniques were employed for this study to collect data from students and lecturers teaching and learning computer programming at different levels of their computer science education.

Computer programming was chosen because the researcher believes that integrating activity-based learning in computer science education is paramount for technological innovation, skills development and employability of the students. Also, the researcher believes that using the activity-based learning approach within computing sciences programming instruction can help to establish a foundation to employ the same teaching approaches in other disciplines.

The responses from students and lecturers were collected from face-to-face interviews, observations, online surveys, questionnaires and focus group discussions within the AR design process. In addition to the primary data, the secondary data were collected through empirical literature studies. Interviews were conducted with lecturers and students at HEI-1, HEI-2, and HEI-3. The semi-structured interview formulation approach using open-ended questions was used to capture data from the students and teachers for each semester. Thus, to employ some level of flexibility in the responses to gather the trends of feedback from the respondents to make informed meaning to the design of the activity-based programming instruction framework.

The quantitative aspect of the study administered questionnaires to the respondents based on the research questions to gather the quantitative data with the AR design process. Focus group discussions among students were recorded and interpreted accordingly to gather data from a group or collective perspective experiences on activity-based learning approaches. The Statistical Package for Social Scientists (SPSS) Version 26 and AMOS 26 using the Structural Equation Model (SEM) were used to analyse the quantitative data while Atlas Ti. was used to conduct thematic analysis for the qualitative data. All the data received and analysed were reliable since the Cronbach Alpha value was greater than 0.7 after reliability and validity tests were conducted. Consequently, the data received aided in the development of the activity-based learning framework for instructing computer programming within a blended learning environment.

1.6 Research Contribution

The contributions of this research to the body of knowledge includes the following;

1. Development of a curriculum mediation based on pedagogic approaches that could support activity-based learning for instructing computer programming in HEIs in Ghana.

2. Establishment of the teaching and learning gains of an activity-based computer programming instructional approach in a blended learning environment from the perspective of students and lecturers of HEIs in Ghana.
3. A new framework that could be used in teaching computer programming using an activity-based learning approach in a blended learning environment of HEIs.

1.7 Thesis Structure

The study covers in total seven chapters. The first chapter provides the introductory part on the background of activity-based learning approach in a blended environment for instructing computer programming. Outline of the problem statement, defining the research objectives, establishing research questions, the scope of the research, a brief methodology for the study and contributions for the study are thoroughly discussed in this chapter.

The second chapter of the study discusses a review of literature and contributions from existing studies. The chapter also provides identifications of research gaps where necessary. Literature on activity-based learning, blended learning, teaching pedagogies in computer science education, smart learning, flip learning and other contributions to the study are reviewed. Also, the chapter provides a review of existing models for the blended learning concept underpinning the research.

The third chapter provides the conceptual framework, philosophical and learning theories used in computer science education.

The fourth chapter discusses the adopted methodology employed for this study towards the development of the ABL framework. This chapter also discusses the methods that contribute to action research designs. In other words, the research design approach, research method, population, sample frame, sample size, validity and reliability instruments, etc. are defined in this chapter. The phases and timeframes that need to be accomplished within the stipulated time period are also defined in the fourth chapter.

The fifth chapter of the study presents the findings and results from the study. The data from the first and second semester of the action research is analysed and presented. The chapter also presents the design of a curriculum mediation based on pedagogic approaches that supports activity-based learning for instructing computer programming in HEIs in Ghana.

The sixth chapter of the research discusses the findings from the third semester of the AR and the development of the framework for activity-based computer programming instruction with a blended learning environment.

Finally, the seventh chapter of the study presents the discussions of major contributions to the body of knowledge based on policy, theory and practice. The summary of all the research findings, recommendations and suggestions for future research and the conclusion of the research is presented in the seventh chapter.

1.9 Conclusion

This chapter has outlined the background and the problems related to the research gaps of teaching and learning of computer programming in HEIs within a blended learning environment. It is noted that key problems with most HEIs in undergraduate computer science academic programmes are how to teach programming and how students can comprehend and write codes or programs. The scope of the study focused on teaching and learning of computer programming in HEIs within a blended learning environment of three HEIs in Ghana.

The researcher has detailed the key research objectives and research questions that need to be answered upon completion of this study to avert the challenges that most HEIs face in computer programming instruction. The action research approach was used to conduct the study which spanned within three semester periods. A brief details of data collection using the mixed method approach was briefly explained in this chapter. The contributions for this study and the entire structure of the study have also been discussed in this chapter.

CHAPTER 2

REVIEW OF LITERATURE

2.1 Introduction

In this chapter, a review of existing literature is conducted and discussed in the area of activity-based learning approaches in the instruction of computer programming in the scope of a typical blended learning environment. The driving force for this study is to develop a framework for an activity-based approach for instructing computer programming within a Blended Learning (BL) environment. In view of the fact that the BL environment is used for this study, a review on the BL approach in teaching and learning was also conducted.

The chapter presents the reviewed computer programming instructional approaches in HEIs. Consequently, the teaching and learning strategies of computer programming instruction were reviewed and criticised. Likewise, the challenges in the teaching and learning of computer programming were reviewed in this chapter.

The reviewed literature also averred that teaching and learning programming remains a difficult task among students and teachers. This is because most students face a lot of comprehension difficulties in their programming classes. The literature also shed light on the need to enhance students' cognitive and critical thinking abilities in programming.

2.2 Educational Paradigms in HEIs.

Conner (2020) explained learning as an act, process, method, or personal experience of acquiring skills or knowledge. In the same way, an effective learning experience requires one to be taught, hence the approach of helping an individual to acquire skills, knowledge, experience and competences is defined as teaching (Alkhatib, 2018). This presupposes that teaching and learning is an interactive approach, thus requiring two key personalities (a teacher or an instructor and a learner) in an attempt to achieve the desired educational results formally or informally (Moorthy and Arulsamy, 2014). The essence of learning is to make changes to society and behaviour (Alkhatib, 2018).

Studies have confirmed that over the past two decades, HEIs have gone through a global transformational shift under different important factors (Chung, Subramaniam and Dass, 2020; Dhawan, 2020; Moorthy and Arulsamy, 2014; Alkhatib, 2018) especially in the recent Covid-19 pandemic. These factors include the emerging influences of Internet technologies and associated teaching and learning tools which have brought creativity in the delivery of HEIs academic programmes and meeting the needs of quality teaching and learning experience (Garrison, Cleveland-Innes and Fung 2004; and Kundi and Nawaz, 2010). Consequently, it can be underscored that the use of Internet technology in HEIs has brought a paradigm shift in three major educational adoptions (Sasseville, 2004, and Alkhatib, 2018). These are:

1. Management of institutional information resources (courses, e-resources, e-service, online registrations, etc.)
2. Course delivery (Online course management systems, Moodle, Sakai, Blackboard, Smart Board, etc.)
3. Adoption of third-party interactive tools and e-applications (Turnitin, Screencast-o-matic, WebCT, etc.).

There have been tremendous advancements in the use of digital devices used in HEIs for teaching and learning. Among these are smartphones, iPhone, iPad, laptops, etc. These allow seamless access to learning resources anytime and anywhere provided one has Internet access. This affirms the studies of Cook and Sonnenburg (2014); Hwang (2014); Hwang et.al (2015); Spector (2014); Gros (2016) and Alkhatib (2018) on the relevance of reforms in HEIs.

Nel (2017) asserted that higher education instructors must modify their pedagogical techniques in the face of multiple obstacles in HEIs in order to provide students with meaningful and engaged learning experiences that are likely to enhance student success and effectively educate them for the world we live in. Nel (2017); Gros (2016); and Alkhatib (2018) equally affirmed that instructors must evaluate the potential of digital technology to help flexible pedagogies, as well as the role that students might play as participants in the learning process, as part of the pedagogical change.

Also, it can be underscored clearly that the advent of technology in the world has forced most HEIs to adapt to the change. It is noted that HEIs in recent times are compelled to acclimatise to the inclusion of digital culture and the digital world (Blanco et al. 2020; Chung, Subramaniam and Dass, 2020; Dhawan, 2020; Alkhatib, 2018; QAA, 2018; Chew, Jones and Turner, 2008). Anything contrary to this means that such HEIs are very likely to perish. HEIs are known to be a hub to produce intellectuals that solve world problems, hence they are known to be an intellectual society where knowledge is produced. In view of that it is absolutely intolerable to grow such intellectual communities without the use of technology in the present world.

The technological advancement in the digital world has pressurized HEIs to promote technology in the frontiers of teaching and learning (Blanco et al. 2020; Chung, Subramaniam and Dass, 2020; Dhawan, 2020; Chung, Subramaniam and Dass, 2020; Dhawan, 2020). Also, countries like the United States (US) and the United Kingdom (UK) have demonstrated a keen footstep towards educational developments such as the “one laptop per student” and open and online degrees to make higher education accessible to all citizens (MTH, 2007; OLPC, 2008; OU, 2008, Warschauer, Cotton and Ames, 2012). Moreover, the present generation of learners' intellectual curiosity has necessitated the implementation of new creative teaching and learning approaches by educators and institutions (Mavuso and Jere, 2020).

Nonetheless, the pressure from the digital societies to integrate technology and e-learning in some HEIs faced massive failures as a result of several factors such as pedagogical consideration and challenges, cultural differences, different economic factors, etc. (BBC, 2006; Meyer, 2006). To unravel these phenomena of the paradigm shift in HEIs to integrate technology in their digital societies, Turban et al., (2002) underscored the following three pressures among HEIs. These are pressures in technology, globalisation, and society. Figure 2.1 demonstrates the demands of higher education.

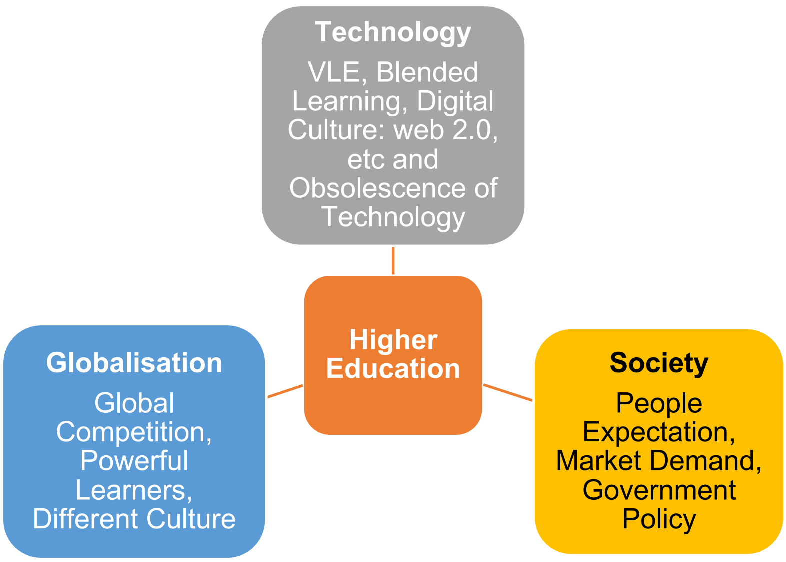


Figure 2.1: The Demands on Higher Education Amended (Turban, 2002)

The persistent pressure to integrate technology in HEIs as indicated in Figure 2.1 amounts to a disturbing phenomenon to transform higher education's mission, vision and policies. The global competitions and diversifying socio-cultural expectation of the digital society have driven HEIs from being "self-sufficient" institutions to a "knowledge" creation hub for teaching and learning (Baty, 2006; Turban, 2002; Nel, 2017; Mavuso and Jere, 2020; Vázquez Cano and Gisbert, 2015; Olsson and Mozelius, 2016; Klašnja-Milićević et al., 2017).

The assessment practices in HEIs have also undergone a dramatic change from the traditional method of assessment to a pragmatic approach using technology which enhances academic operations and performance. It is obvious that the use of technology has aided in the assessment practices and granting of quality feedback to students (Rochefort, 2011 and Alkhatib, 2018). These assessment processes aid HEIs to demonstrate effective learning gains, achievements and drawbacks in the pedagogical methods employed. Hence, it is noted from literature that there has been a transformational change in the HEIs to integrate e-learning for teaching and learning.

Klašnja-Milićević et al. (2017) developed an e-learning system called "Programming Tutoring System, Protus" to enhance teaching and learning of Java programming in a HEI. The system architecture of the Protus system as indicated in Figure 2.2 indicates that to enhance teaching and learning in HEIs, it requires the need for establishing a system that links the students and the teachers. The teacher provides instructional materials, views learning contents and reports upon access while the student accesses the online portal via a web browser to conduct tests, submit assessments, view tutorials, and communicate with the teacher, peers, etc. upon an authentication access.

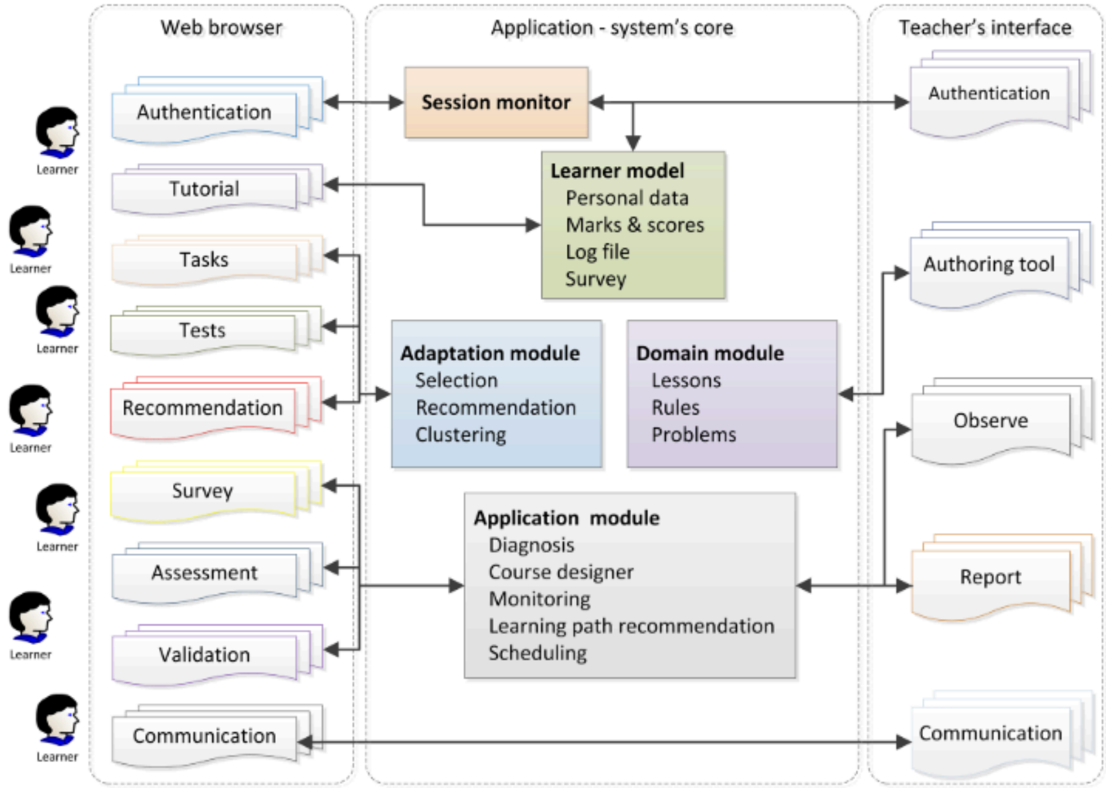


Figure 2.2 E-learning System Architecture (Protus) Derived from Milićević et al. (2017)

Applying the Protus framework in the context of computer science education delivery seems to be a very good approach for teaching and learning. However, the model is seen to be too abstract and generalized for teaching and learning in HEIs. Hence, applying the Protus model in computer science education with emphasis in computer programming instruction especially in Africa and the Ghanaian context where most students lack basic IT skills, have weak computing skills, have poor problem-solving approaches, etc. may not be fully applicable. Hence, the need for this study to fill those gaps to enhance the cognitive, social and teaching capabilities of both students and teachers. This is in line with the study of Kintu, Zhu and Kagambe (2017) which finds that there is a need to promote an activity-based approach to enhance cognitive development among students within a blended learning environment for computer science education.

2.3 Computer Science Education

Computer Science (CS) as an academic discipline scientifically studies "*algorithmic processes that describe and transform information: their theory, analysis, design, efficiency, implementation, and application*" (Denning et al., 1989, p.12). Teaching and learning in the field of computer science has for the past four decades been seeing a significant growth (Sharmin, 2020, 2018; Ahadi, Vihavainen and Lister, 2016). It is worthwhile to know that CS is not all about programming however, programming is seen as the most acute CS fundamental skill (Denning et al., 1989; The Joint Task Force on Computing Curricula, 2013). Studies of Gal-Ezer and Harel (1998); Lye and Koh (2014); Wing (2006, 2008) assert that computer programming is the foundation to learn Computer Science theories, principles and concepts.

According to Alharbi and Hannaford (2011) extant literature in the Science, Technology, Engineering and Mathematics (STEM) education reveals that computer science as a subject is relatively new and as such is quite limited when compared with other disciplines of science such as physics and chemistry. However, the need to continue in computer science education is now more important than ever due to both societal and economic benefits it brings. A key motivating factor for computer science education especially at the undergraduate level, is the resolving of high failure rates and low retention rates among students who pursue computer science courses in different higher institutions of learning (Ahadi, Vihavainen and Lister, 2016; Bennedsen and Caspersen, 2007; Kinnunen and Malmi, 2006; Lister et al., 2007; Watson, 2014; Zingaro, 2014).

Some of the factors identified in literature as contributing to withdrawal from computing science educational programmes (i.e., computer science, information technology, software engineering, information systems and computer engineering) include lecture attendance and preference for working alone (Simon et al*.,* 2012; Sheard et al., 2011), work overload (Rountree et al., 2004), students' self-efficacy and confidence in their ability to program (Lewis and Loftus, 2009) and perceived value of computer science to solving real world issues (Biggers, Brauer, and Yilmaz, 2008).

In a different study Lang et al. (2007) discovered that factors that influence the rate of retention in computer sciences educational programmes include gender factors in which females are more affected and pedagogical factors which are associated with or related to curriculum and assessment. Petersen et al. (2016) and Kinnunen and Malmi (2006) opine that lack of time and motivation, poor time management skills and perceived difficulty of computer science as a course contribute to the poor performance and lack of interest in computing sciences educational programmes among students.

To foster teaching and learning and motivation among students in the field of computer science a number of innovative teaching tools and methods have been used by instructors. Some of the tools employed include, peer-led team learning (Horwitz et al., 2009), project-based learning (Haungs, Clark, Clements, and Janzen, 2012), lab-centric teaching (Titterton, Lewis, and Clancy, 2010) and interactive multimedia (Blank, Roy, Sahasrabudhe, Pottenger, and Kessler, 2003). According to Haungs et al., (2012) performance of students in computer science can be enhanced when an assignment is personalized based on the different interest-based tracks of students. In a similar study Alhazmi et al. (2018) examined how assignment choices of students can enhance motivation in computer science. This according to VanDeGrift (2007) inspires creativity as students carry out more open-ended assignments.

Extant literature in the field of computer science education indicated that students' prior experience and gender have received the attention of several scholars. In relation to prior experience, computer science education emphasizes that the prior experience of students to computing such as to coding and programming help them to perform better than those who are first timers in the course (Hagan and Markham, 2000; Wilcox and Lionelle, 2018). In a related study Tafliovich et al. (2013) confirmed that students perceive that prior experience is necessary for success in computer science.

The computer science and STEM fields are generally recognized as gender biased. It is for this reason that prominent people such as Bill Gates, Mark Zuckerberg and Barack Obama are seriously pushing for coding to be part of the reading and writing in schools (Code.org, 2016). As NY Times writer Tiku (2014) states however, "*if coding is the new lingua franca, literacy rates for girls are dropping*". According to the National Science Board (NSB) (2016), among all students enrolled in computer science, 81% are males while only 19% are females in American secondary schools. With the current dominance of men in the computer science world with little consideration to gender diversity it is imperative to engage and involve women in computer science (Thomson, 2016). Thus, encouraging female students to learn computer science would empower women to also take a position in this profitable and powerful field in addition to providing the benefit of diversity for the purpose of enhancing technological development (Margolis and Fisher, 2002).

2.4 The Concept of Computer Programming

Computer Programming is a process of developing or writing sets of instructions to execute a certain task or problem. The sets of instructions are termed as the source code which are written in one or more programming language to solve a real-problem (Kasame, Pachoen and Manit, 2016).

Computer programming entails various tasks which includes: analysis, developing algorithms, developing algorithm's accurateness, implementation of the algorithm based on a selected programming language that are understood by the programmer in a high-level language rather than machine language which is executed by a central processing unit. The relevance of computer programming is to find a set of instructions that automates the performance of a task to solve a real-world problem. Expert programming accordingly mostly requires knowledge in numerous different areas, which include knowledge of the programming language, understanding of the programming domain and problem, algorithms, and logics (Kasame, Pachoen and Manit, 2016; López-Cruz et al., 2017).

The first computer program was published in 1843 by mathematician Ada Lovelace, who devised an algorithm to produce a sequence of Bernoulli numbers for Charles Babbage's Analytical Engine (Fuegi and Francis, 2003). External punched cards were originally used to store data and instructions, which were preserved in order and grouped in program decks. Herman Hollerith introduced the notion of storing data in a machine-readable format in the 1880s.
(da Cruz, 2020). Later, he added a control panel (plug board) to his 1906 Type I Tabulator, which allowed it to be programmed for different duties, and by the late 1940s, unit record equipment like the IBM 602 and IBM 604 were programmed in the same way. Nonetheless, with the introduction of the idea of the stored-program computer in 1949, both programs and data were stored and manipulated in computer memory in the same approach to solve a problem.

In recent times, the problem solving approach requires proper requirement engineering and design which is seen in the conventional Software Development Life Cycle (SDLC) (Techbeamers, 2019; López-Cruz et al., 2017). In software systems development or engineering, the SDLC is a process of planning, analysing, designing, building, and testing. It is also worth to know that before one can develop a system, the developer will have to write a series of codes, this act is known as programming. Before a system can be developed one needs to use a programming language. Examples of programming languages are C++, C, Java, C#, VB.net, ASP.net, Python, PHP, Adriano, HTML, etc. (López-Cruz et al., 2017). These languages are known to be high-level programming languages. Consequently, before a software application is executed, the program is first written in a high-level programming language that is understood by the developer and translated to a low-level machine language through a compiler. The compiler then converts the high-level language to a low-level machine language in 0's and 1's for the computer to execute the program.

The researcher sees the SDLC as the bedrock for developing a good software program and for that matter students must be taught how to follow the cycle to develop a program. The SDLC as indicated in Figure 2.3 constitutes a holistic development of a computer program or software development by planning and identifying the problem of a system, analysing the problem, designing and developing an appropriate algorithm or the core specification of the problem requirements, building the actual system by coding and testing, implementing the design and testing for errors, deploying the system, evaluating the system and maintaining the system. It is assumed that this classical approach used in developing software applications covers all aspects of system design from initiation and the problem identification stage to execution and the maintenance stages of the system (Techbeamers, 2019; López-Cruz et al., 2017).

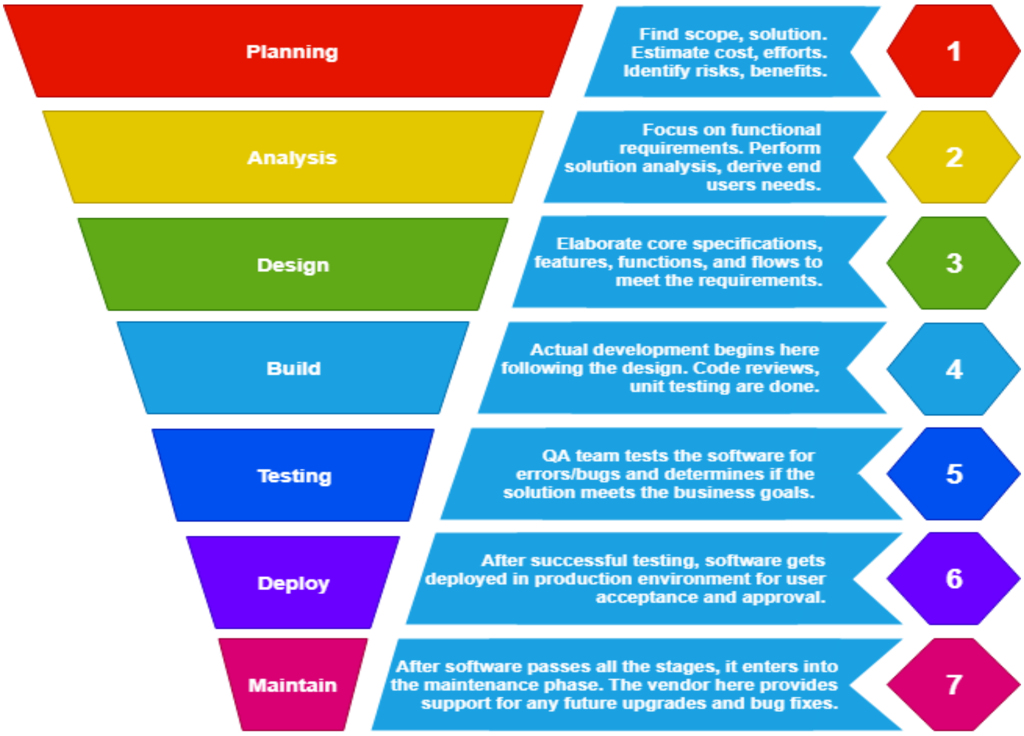


Figure 2.3 SDLC, Summary by Techbeamers.com, (2019)

The first and second phases of the SDLC require planning and analysis of the problem of a case study or a real-world scenario. This implies that the programmer must comprehend and read the necessary requirements needed to solve the problem. After the problem identification and requirement analysis, the problem is then divided into lesser tasks to decipher possible solutions to the problem in the next phase by designing core specifications, flow charts, algorithms that define the step-by-step approaches to the problem, etc.

The fourth phase picks up the decomposed problems and develops the real codes that perform the tasks of the defined problem. Unit testing is conducted on the design. Also, the fifth phase tests the design to determine if there are any bugs to be debugged. Debugging is the process of fixing errors or bugs in the codes and establishing that the software is performing what was defined in the algorithm. Li and Wang (2013) asserts that the purpose of debugging a programming source code is to establish if there are no syntax errors or logical errors. Finally, the system is deployed and maintained to solve the intended problem that needs a solution.

Garner (2003) asserts that the traditional SDLC should be fundamental to all novice programmers, in this case undergraduate students who are learning how to program and must master the phases religiously to enhance their programming skills. Greyling et al. (2006) correspondingly posit that it is very relevant for students to thoroughly understand the problem of a question or case study before attempting to write the source code thus, the need to equip the teacher with correct pedagogic approaches for teaching programming.

2.5 Teaching and Learning of Computer Programming

Teaching and learning of computer programming are quite different from all other conventional courses like business, social sciences, agriculture, etc. based on my personal teaching and research experience for the past decade. This is because the approach of teaching programming requires a systematic approach for solving real-world problems by programmers. Additionally, the demand for the global transformation and efficiencies to execute business activities has been the clarion call for most HEIs to design curriculums that can meet such demands (Kinshuk et al., 2016; Wilcox and Lionelle, 2018).

Consequently, there is the need for HEIs that run computing degrees to introduce computer programming languages that can equip learners to apply the knowledge learnt to transform the world through top-notch software applications. Stemming from such motivations by universities, it is then very necessary to select an appropriate technique that can be used for the instruction of programming courses (Ala-Mutka, 2003; Ahadi, Vihavainen and Lister, 2016; Bennedsen and Caspersen, 2007; Kinnunen and Malmi, 2006; Lister et al., 2004; Watson, 2014; Zingaro, 2014). It is noted from existing studies and practice that teaching and learning of computer programming still possess challenge among students (López-Cruz et al., 2017; Lister et al., 2007).

In the next sections, the researcher discussed the problems of teaching and learning of computer programming, the reasons why learning computer programming is difficult and how the problem is addressed for the comprehension of computer programming syntax and semantics.

2.5.1 Problems of Teaching and Learning of Computer Programming

In the studies of Greyling et al. (2006); Ahadi, Vihavainen and Lister (2016); Watson (2014); Zingaro (2014) three identifiable problems that most teachers face when teaching students to code or program via a text-based computer programming language in HEIs are;

1. Too much focus on writing syntax correctly. This approach mostly compels students to understand the programming syntax by all means. Hence, students at early programming stages begin to get the misconception that programming is all about writing correct syntax codes that conform to the rules of the programming language and the compiler. This misconception turns out to ruin their programming experiences since most students get demotivated with a lot of syntax errors (Lister et al., 2004; Greyling et al., 2006; Ahadi, Vihavainen and Lister, 2016).

2. Also, the problem-solving phase which is very critical in the software development is also least to be desired, since little effort is mostly devoted by teachers on the problem-solving phases of system development (Lister et al, 2007; Greyling et al., 2006; López-Cruz et al., 2017; Lye and Koh, 2014).

3. Finally, one critical factor resulting in the poor programming skills of most students is the lack of learner support in the development phase of constructing meaning on the problem. In most cases after the teacher teaches the student, the students are unable to gain support to advance in the comprehension of the syntax and the logics of the codes. It is noted that teachers mostly concentrate on the syntax rather than the semantics, hence resulting in poor execution of codes with logical inaccuracies in the program as averred in the studies of Greyling et al. (2006); Ahadi, Vihavainen and Lister (2016); Watson (2014) and Zingaro (2014).

2.5.2 Reasons Why Learning Computer Programming is a Problem among HEIs Students

Learning to program is a human driven activity which is mostly inspired by individual intrinsic factors such as self-confidence, self-motivation, level of satisfaction and comfort (Smith and Ungar, 1995; Robins et al., 2003). In view of that students do not only need the comprehension of the prevailing syntax and semantics of the programming language but will need an individual self-readiness and motivation to pursue programming courses.

According to Wiedenbeck, LaBelle and Kain (2003) learning programming goes beyond just acquiring the skills to building on the students' cognitive skills and self-efficacy to solve real-world problems. Ahadi, Vihavainen and Lister (2016), Wiedenbeck, LaBelle and Kain (2003) further aver that learning programming requires students' efforts, different levels of coping tactics, tenacity in times of disappointment, and individual goals and visions. Hence, computer programming provides that level of challenge to most students since students have different learning styles and tenacities to comprehend programming knowledge and skills.

Besides, learning computer programming requires the application of convoluted cognitive abilities of the student which consequently support the student to reason and solve real-world problems (Tie and Umar, 2010). Computer programming constitutes different logical structures which translate to a machine language that a compiler can understand and interpret to a human readable language or high-level language (Wiedenbeck, LaBelle, and Kain, (2003), yet its comprehension level is quite difficult among most students. Hence, most students easily get frustrated in handling the bugs that emanate from the syntax and logical sequences of the programming language and perceive that getting the syntax wrong means the end to pursue further (Vogts et al., 2010).

Ismail et al. (2010) assert that the prevailing challenges of most students in computer programming classes are that students have got deficiencies in meta-cognitive know-how to solve programming questions and concepts. Ismail et al., (2010) indicate that the situation is not only common in South African universities but it has remained a global challenge especially in Sub-Sahara Africa. Other researchers such as Scott et al. (2011, p.336) also indicate that the prevailing challenges in programming arise as a result of poor high school educational background. Hence, they assert the need for developing different strategies to enhance teaching and learning right from the high schools.

Lister et al. (2004), Robins et al. (2003), Ahadi, Vihavainen and Lister (2016) and Vogts et al. (2010) put it straight to the point that learning to program is mostly difficult and has got an alarming failure rate. Robins et al. (2003) add that the situation is even worse among novice programming students. Butler and Morgan (2007) conducted a study on students who were enrolled on a foundational programming class at a HEI in Australia. Analysis on their study concluded that there was a relatively high level of failure rate among the first-year students in the programming class. Their students lamented and perceived programming as the worst and most boring course among their enrolled courses. This negative mind-set turned out to become a hindrance to tackling advanced programming courses with positive cognitive reasoning.

Also, previous studies have shown that students with poor mathematical background usually find it difficult to comprehend programming constructs and logics to solve programming questions (Mayer et al., 1989; Byrne and Lyons, 2001; Butler and Morgan, 2007, Rogerson and Scott, 2010). Byrne and Lyons, (2001) affirm that mathematical sciences have direct relations with computer programming hence, a lack in mathematical knowledge hampers comprehension levels towards programming among students.

Moreover, the external influence to reform higher education in computer science disciplines forces most HEIs to run professional programming courses which in most cases lack pedagogical support (Vogts et al., 2010). They contend that the drive to introduce programming in a computer science curriculum is a good idea especially using a different development environment which is relevant to the industry.

Kanaparan et al. (2013) assert that the difficulties in programming usually result in two different types which include both cognitive and behavioural characteristics of the student. Their study stressed that these two broad challenges faced by students hinders students' capabilities for computer programming skills. On the other hand, Kanaparan et al. (2013) aver that these challenges can be resolved by the use of different technology driven instructional approaches. The technology driven approaches are known to enhance both cognitive and behavioural skills using activity-based approaches.

In addition to the above, Yacob and Saman (2012) also acknowledge the fact that students face a lot of programming challenges which prevent them from building competence and proficiencies to program. Yacob and Saman conclude that the high rate of student failure in programming courses can be unravelled by motivating the students and engaging them practically through activities.

Al-Imamy and Alizadeh (2006) assert that teaching programming requires three different pedagogic goals which include:

1. the understanding of the programming language syntax,
2. enhancing the development skills and
3. provisioning of problem-solving and creativity skills.

Al-Imamy and Alizadeh aver that these three pedagogic goals also pose challenges to the teachers who teach the programming course. The researcher believes that having a qualification to teach at the higher education as a computing science lecturer does not guarantee the instructor or a lecturer to possess the skills of quality teaching of programming in HEIs.

Finally, the researcher on the other hand cannot only attribute the above challenges in teaching and learning of computer programming to pedagogic failures, behavioural changes and cognitive shortfalls. Thus, the researcher agrees with previous studies such as Greyling et al. (2006); Tie and Umar, (2010); Sarpong, Arthur and Amoako (2013), etc, who assert that most students usually enter HEIs "under-prepared". For example, Greyling et al. (2006) assert that most South African students who enter HEIs for computing disciplines find it very difficult to comprehend foundational courses that employ computing devices as pedagogic aids. Hence, most students find it difficult to use a computer to write basic programming codes. In the same vein, Sarpong, Arthur and Amoako (2013) who conducted their research in Ghana also affirmed that the calibre of students who get enrolled into computing programs lack basic skills to read computing degree programmes such as computer science, information technology, and information systems. They assert that students who enroll into computing programmes experience challenges in problem-solving and basic critical thinking skills in programming classes.

2.5.3 Addressing the Problems in Teaching and Learning of Computer Programming among HEIs Students

Computer programming as discussed involves all aspects of software development from identifying a problem, writing a source code, testing for bugs and debugging, evaluating and maintenance of a software or a source code (Saeli et al., 2011). A lot of researchers have vehemently asserted that teaching and learning of programming in HEIs has always been a herculean task (Ahadi, Vihavainen and Lister, 2016; Ala-Mutka 2003; Overmars, 2003; Ali, 2005; Linn, 2009; Saeli et al., 2011; Sarpong, Arthur and Amoako, 2013).

As a result of that numerous researchers and educational advocates such as Nouri, (2016); Kintu, Zhu and Kagambe, (2017); Nel (2017); Wulf, (2005); Margaryan, Collis and Cooke, (2003); and Fisher, Byrne and Tangney, (2016) have underscored the need for reforms in HEIs in terms of teaching and learning approaches.

Saeli et al. (2011) asserted that students have different comprehension levels of programming and thus, alluded to the fact that some students who even take as far as two years of tuition in programming classes still find it difficult to learn and develop their own programming skills. Hence, Saeli et al. made recommendations for reforms in computer programming curriculum and instruction. The researcher strongly believes that learning to comprehend and develop programming skills can be achieved and mastered if the teaching and learning approaches are conducive and suitable. This is alluding to the educational reform agenda by Nouri, (2016); Kintu, Zhu and Kagambe, (2017); Nel and Wilkinson (2006); and Saeli et al. (2011).

According to the studies of Ahadi, Vihavainen and Lister (2016) and Ala-Mutka (2003), teaching and learning of programming is a difficult task, hence they recommended that instructors of programming classes must design appropriate teaching and learning strategies to avert the challenging phenomena. In addition to that Ala-Mutka (2003); Ali (2005); Linn (2009); Saeli et al., (2011); Sarpong, Arthur and Amoako (2013) affirm that most students at their early programming journey seem not to comprehend the basic programming concepts, therefore developing a hands-on approach, practical activities, case studies, etc. increases the comprehension level of students in first year programming classes in HEIs.

As a result of this, different strategies need to be employed to promote and enhance the abysmal performance of programming students. Robins et al. (2003) posit that to avoid the challenging situation of teaching programming, teachers must adopt different strategies by combining different features of teaching concepts to enhance students' understanding. They assert for instance, that teachers must build "visualized examples" using different scenarios to enhance understanding of diverse programming problem solving strategies. Consequently, Robins et al. (2003) assert that the visualized examples will develop a cognitive "library" for the student to recognize and apply the same strategy for solving different problems.

Also, Abbotts et al. (2008) indicate that there is a need to involve students and teachers to establish learning in practically orientated approaches to enhance teaching and learning using the constructivist approach (Bruner, 1961) to guide a learning process. Abbotts et al. (2008) posits that a good practical approach of teaching and learning will aid to build a good learning experience among students. Additionally, Merrill (2003) affirms that learning is effective when the learner is involved in practical problem solving of a real-world situation. These principles are relevant to the field of computer programming instruction where the learners are mostly novices in programming languages, and so different strategies must be adopted to foster programming instruction.

It is on this note that the researcher took interest to add to the body of knowledge an enhanced approach using the activity-based instructional approach to teach programming within a blended learning environment.

2.6 Instructional Approaches in Teaching Computer Programming

In this section, the researcher identifies different teaching strategies of computer programming that enhance students' comprehension, among these instructional strategies are, problem-based learning, puzzle-based learning approach, pair-programming, and game-themed approach.

2.6.1 Problem-based Learning Approach (PBL)

PBL is a student-centered approach in which students learn about a subject by solving an open-ended question in trigger material. The PBL approach does not emphasize problem solving with a predetermined answer, but it does allow for the development of other desired abilities and characteristics. This involves improved group cooperation and communication, as well as knowledge development. The PBL approach was created for medical education and has now been expanded to include other learning programs. Learners are able to gain abilities that will be useful in their future practice as a result of this procedure. It improves critical thinking, literature retrieval, and supports on-the-job learning (Schmidt, Rotgans and Yew, 2011).

In the 1960s, Barrows and Tamblyn developed the PBL method at McMaster University in Hamilton's medical school curriculum. Traditional medical education disillusioned students, who saw the huge quantity of material taught in the first three years of medical school as having little relevance to practical care (Barrows, 1996). The PBL curriculum was created to encourage students to study by helping them to understand the connection and applicability of what they are learning to future responsibilities. It keeps students more motivated to learn and emphasizes the necessity of responsible, professional attitudes and collaborative principles (Barrows, 1996; Cripps, Jacobs, and MacCallum (2020).

The central focus of science oriented academic disciplines is to instruct students to gain problem-solving skills (Docktor, 2009). Several researchers have alluded to the fact that problem-based learning (PBL) is a novel approach towards teaching and learning, since it provides the learners the skills to solve real-world problems (Docktor, 2009; K. Heller and Heller, 2010; Hazzan, Ragonis and Lipidot, 2020). These researchers indicate in the affirmative that the problem-based approach of instructing students enhances students' academic performances and retention.

In addition, a student's capability to solve a given problem involves the student's creative skills that are not only linked to the procedure given to the student (Bolton and Ross, 1997). Docktor (2009) asserted that a novice problem solver can become an experienced problem solver through constant practice of solving real-world problems. Correspondingly, Nuutila, Torma and Malmi (2005) and Wu (2006) assert that PBL emphasises effective "student engagement" in problem-solving. The driving force for the students' engagement in problem-solving is that most students after graduating will be employed to solve professional and real-world problems in their daily professional duties (Nel, 2017). The PBL approach enhances a greater level of critical thinking, excellent orientation to knowledge in software or system development, and hands-on skills development among students involved in problem solving activities (Hazzan, Ragonis and Lipidot; An, 2013; Myrup et al., 2017)

According to An (2013, p.1) "*The current information age society needs people who can think critically and creatively and can effectively use ever-increasing amounts of data to solve ill-structured problems, to make decisions in the face of uncertainty, and to collaborate with other people*".

The assertion by An (2013, p.1) is directly linked to an effective engagement of students using PBL approaches in teaching computer programming. Moreover, the "International Society for Technology in Education (ISTE)" issued a stringent standard in HEIs dubbed the "International Society for Technology in Education Standard" which provides quality in HEIs for learners or students, lecturers or teachers or facilitators and administrators or managers (ISTE, 2020). The ISTE standard for students or learners specifically, consists of "*creativity and innovation, communication and collaboration, research and information fluency, critical thinking, problem-*

solving, and decision making, digital citizenship, and technology operations and concepts" (ISTE, 2020).

Nuutila, Torma and Malmi (2005) identify seven different steps that enhance students' learning capabilities in computer programming. These include engaging students to share their prior knowledge on the topic, providing hands-on practical activity to solve a problem, requiring students to elaborate and share what they have learnt, engaging in group discussions to brainstorm on the problem, identifying and solving real-world problems, and establishing the new knowledge they have acquired and the new knowledge they still need to know.

Nuutila, Torma and Malmi further indicate that students continue their PBL journey by aligning their PBL activities to the learning outcome or goals of the topic and finally submit the solutions of the given problem to their instructor or teacher. An example of PBL activity given to students by their instructors is demonstrated in Figure 2.4.

You have been approached by the owner of a fast-food restaurant to automate the operations in the restaurant. The following menu is offered by the restaurant.

No.	Item	Price	Remark
1		RM 8.00	1 burger/chicken; 1 drink (medium); 1 French fries (medium)
2	Breakfast Set	RM 5.00	1 breakfast; 1 drink (medium)
3	Kiddies Set	RM 6.00	1 burger/chicken; 1 drink (small); 1 French fries (small)
4	Promotion	RM 3.00	Burger/chicken
5	Drink	RM 2.00	Medium size
6	French Fries	RM 3.00	Medium size
7	Dessert	RM 3.50	Ice cream/pie/cake
8	Upgrade meal	RM 1.00	Small to medium / medium to large

- The above price is not inclusive of the 5% government tax.
- An additional 10% will be charged for those dining in the restaurant where 5% will be charged for take-aways.
- An additional 3 % will be charged for payment with credit cards.

The owner has requested that you develop the application with the following requirements:

- Staff can login into the system.
- Staffs are able to check the price for each item.
- Customer order are displayed
- Bill calculation and printing.

Figure 2.4 Example of Problem-based Learning in Programming

From the above problems in Figure 2.4, students are tasked to provide solutions to the problem for the fast-food restaurant.

Nonetheless, Wu (2006) assert that apart from the PBL approaches indicated by Nuutila, Torma and Malmi (2005), there are other PBL approaches that can be employed in teaching computer programming. For example, Wu (2006) and Myrup et al. (2017) indicate that problems are not only given to students to solve but rather the problems are given to the students as source of motivation to reveal a new theory.

Moreover, researchers have indicated in the affirmative that the problem-based learning approach in computer science education enhances students' academic performances and retention (Wu, 2006; Docktor, 2009; K. Heller and Heller, 2010; Hazzan, Ragonis and Lipidot, 2014) and thus applied it for an activity-based computer programming instruction as used in this study. Also, Nuutila, Torma and Malmi (2005) claim that the PBL approach in computer programming improves students' skills in communication, critical thinking, intrinsic motivation and sense of responsibility. Finally, the researcher strongly believes that the PBL approach is directly linked to the activity-based approach the researcher chose for investigation in this research (Hazzan, Ragonis and Lipidot, 2020).

2.6.2 Puzzle-based Learning Approach

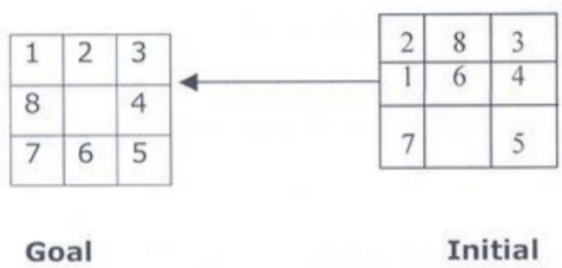
The use of puzzles to develop higher-order thinking abilities like problem-solving is known as puzzle-based learning. Falkner, Sooriamurthi and Michalewicz (2010) asserted that the goal of the puzzle-based learning technique is to get engineering and computer science students to think about how they frame and solve issues that aren't found at the end of a textbook chapter. By presenting a range of puzzles and their solutions. Again, they affirm that puzzle-based learning aids to engage students while also boosting their mathematical awareness and problem-solving abilities. The training is built on the great traditions taught over the last 60 years by Gyorgy Polya and Martin Gardner (Falkner, Sooriamurthi and Michalewicz, 2010).

The ultimate purpose of puzzle-based learning is to provide students with the foundation they need to be competent problem solvers in the real world. The teaching and learning approach using puzzle-based learning (PZBL) is also one of the approaches that aids in the teaching and learning of programming with the aim of enhancing students analytical reasoning and problem-solving skills (Merrick, 2010; Yoneyama et al., 2008; Falkner, Sooriamurthi and Michalewicz, 2010). The rationale for using puzzles is to enable students to remember and produce a result that is not known to them towards answering a problem. Also, puzzles provide to the student a challenging problem which requires the students to answer without any clear solution to the problem, yet anticipating for the assurance of a solution (Falkner, Sooriamurthi and Michalewicz, 2010).

In the teaching and learning of computer programming, the application of puzzles is applied in different approaches. For example, the instructor or lecturer presents to the students the problem of the day that relates to the learning goals of the topic taught for the day. The lecturer then presents a complete written source code that is mixed-up into different segments forming a puzzled source code. The lecturer decides on the level of difficulty of the puzzle. Subsequently, the lecturer presents the puzzle question to the students to attempt and correctly re-construct the program source code to a running code free from syntax and semantic errors. An example is shown in Figure 2.5.

1) 8 Puzzle Problem (A* search algorithm required)

The 8 puzzle consists of eight numbered, movable tiles set in a 3x3 frame. One cell of the frame is always empty thus making it possible to move an adjacent numbered tile into the empty cell. Such a puzzle is illustrated in following diagram.



The program is to change the initial configuration into the goal configuration. A solution to the problem is an appropriate sequence of moves, such as
"blank tile to the right, blank tiles to the up, blank tile to the up, …"
OR
"move tile 5 to the right, move tile 7 to the left, move tile 6 to the down, …"

Figure 2.5 A Puzzle Problem for A* Search Algorithm

Students are then rewarded for a successful resolution to the puzzle problem. The approach is usually guided by a lecturer on an automated application to shuffle the puzzle problem (Yoneyama et al., 2008). Furthermore, Merrick (2010) asserts that students expressed a high level of interest and active participation in PZBL activities in an introductory computer programming class.

2.6.3 Pair Programming Approach

Pair programming is a method of agile software development in which two programmers collaborate at the same time at a single workstation (Laurie, 2001). One is seen as the “driver” and the other seen as the “navigator” (Zacharis, 2011) as demonstrated in Figure 2.6. The driver is seen to be the coder who types the real source codes, whereas the navigator does the observation by checking for possible errors, making inputs with different ideas and recommendations.

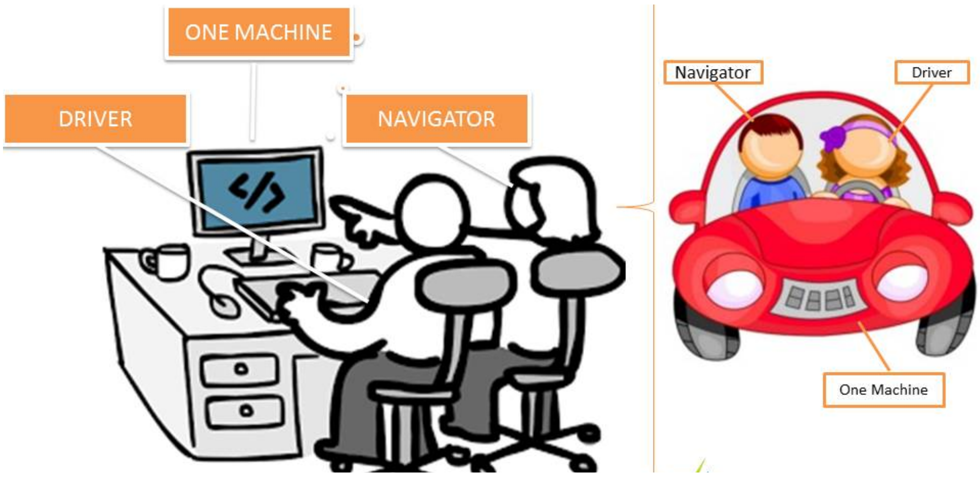


Figure 2.6 Pair Programming Concept (Source: Anarsolutions, 2020)

The observer or navigator evaluates each line of code as it is typed in, while the driver writes code. The two programmers frequently exchange duties. While assessing, the observer thinks about the work's tactical direction, coming up with suggestions for improvements and potential future issues to solve. This is done to allow the driver to concentrate only on the "tactical" aspects of the current task, with the observer acting as a safety net and guidance.

The pair programming approach is also noted to be one of the teaching and learning strategies for computer programming where two programmers in this context (programming students) program simultaneously on a same source code using the same computer (Zacharis, 2011; Chaparro et al. 2005). Both programmers work side-by-side from the initiation of the problem solving, designing, coding, implementing and testing the program.

Previous researchers such as Zacharis (2011), Laurie (2001) assert that the pair programming approach is very effective when developing an application within a constrained time. It provides faster execution time to students than working as individual students on a programming exercise.

Also, the pair programming approach enables students to program with less syntax and semantic errors, enhance their communication skills, build their teamwork spirits, and boost their confidence levels to solve problems.

However, previous studies such as Chaparro et al. (2005) posit that in spite of the underlining positive benefits of paired programming, it can occasionally be infuriating and gruelling. This is because, in most cases students have got different programming abilities and skills, hence pairing a weak student with a good student or vice versa mostly gets irritating to some students. Hence, Chaparro et al. (2005) recommend that great care must be taken when pairing students lest it disadvantages other students.

Another power approach that enhances pair programming is "virtual pair programming" (Zacharis, 2011). Zacharis assert that virtual pair programming eradicates the issues of two programmers sitting at one physical location programming together. This approach uses online tools such as TeamViewer, Zoom remote desktop sharing, etc. to connect the two programmers to program remotely on a project. This approach enables collaborative learning and easy accomplishment of tasks (Cripps, Jacobs, and MacCallum, 2020; Zacharis, 2009).

2.6.4 Pre-recorded Lectures Approach

The process of recording and documenting the material of a lecture, conference, or seminar is known as lecture pre-recording (Castro et al., 2020). It is made up of hardware and software components that work together to record the lecture's audio and visual elements. It is commonly used to give student support in universities and higher education (Castro et al., 2020). According to a 2016 UCISA poll, 71% of universities said this technology was available at their university. When recording lectures on a large scale, the recording system may be coupled with the timetabling system, and metadata gathering can be automated (Smith and Fidge, 2008).

The use of pre-recorded lectures approach (PLA) for computer programming instruction focuses mostly on the use of multimedia learning resources (MMLs) that are accessible to students via an online learning platform (Smith and Fidge, 2008). The MMLs constitute PowerPoint slides that are narrated from the course textbook or lecture notes by the lecturer. The MMLs posted on the online learning platform are usually in video and audio formats. This pre-recorded lecture is given to the students to augment the traditional face-to-face instructional activities of computer programming.

Smith and Fidge (2008) assert that teaching a completed semester course in programming in most cases is very time constrained with time tabling issues. Hence, using the pre-recorded lecture resources helps students who sometimes are unable to absorb the course learning outcomes to comprehend topics where they lack. This approach also works just like the flip learning approach (Nouri, 2016) where students watch pre-recorded video on any topic of their choice, construct their own knowledge and ask the lecturer for support based on what the student has learnt from the pre-recorded lecture.

Contrasting from the orthodox textbook, pre-recorded lectures approaches dichotomize important aspects of the learning outcome where a lecturer highlights salient points from “less important” aspects (Alkhatib, 2018). This approach is noted to guide the students in their programming experience and its illustration practices can be refined anytime by the lecturer based on the students’ learning needs.

Interestingly, the findings of Smith and Fidge (2008) established that lecturers usually perform 5 – 15 minutes pre-recorded lecturers for the students by focusing on different topics based on the course outline. Findings from their research revealed that the pre-recorded lectures approach did not have any significant difference on the final exams score of the students. Smith and Fidge (2008) affirmed in their studies that most students see pre-recorded lecture materials as not compulsory resources for them to watch and listen always. Also, their study revealed that most students do not even access the online resources.

However, in general, their study shed light on the fact that most students responded in the affirmative that the pre-recorded lecture materials enabled them to comprehend certain concepts of their programming course.

2.6.5 Game-based Programming

Gamification of learning is an educational technique that uses video game design and game aspects in learning settings to push students to study (Kapp, 2012; Shatz, 2015). The objective is to increase learners' enjoyment and engagement by captivating their attention and motivating them to continue studying. Gamification, generally defined, is the process of understanding the components that make games entertaining and inspire players to keep playing, and then employing those same components to influence behavior in non-game contexts (Deterding et al., 2011). Gamification, in other terms, is the addition of game features to a non-game environment.

The game-based programming (GBP) approach in teaching and learning computer programming incorporates various interactive graphical user interface applications into the instructional approach (Overmars, 2003; Lasse, 2015; Sung et al., 2010). The rationale for using games in programming is to use the approach to develop different programming concepts via gaming applications. The aim of using the games to enhance teaching and learning of computer programming is not necessarily meant for the students to develop computer games but rather to use the computer games to comprehend programming constructs and concepts (Sung et al., 2010).

Also, the game-based approach provides the students a means to explore and understand logical sequences of programming concepts or different programming constructs in a more graphical approach rather than the classical console interface (Sung et al., 2010). In their studies Tillmann et al. (2013) asserted that teachers can build their own virtual classrooms through games to engage students to program.

In addition, Sung et al. (2010) asserted that GBP assignment can be given to students by firstly, describing the given activity that needs a solution, granting graphical user interface access to the student, and finally alerting students to interact via the graphical interface to program virtually. Findings from Sung et al. (2010) on game-themed programming approach revealed that there was higher level of success in the students' performance than the classical approach of learning programming. The students affirmed that they spent a lot more time using the game-based platform to program than the traditional method of programming on a console interface. Excitingly, most students indicated that the game-based approach empowered their enthusiasm and motivation to program (Sung et al., 2010). An example of game-based learning include the Box Island programming gaming tool as indicated in Figure 2.7

The Box Island: This is an awe-inspiring mobile programming and coding tool that aids programmers on an adventurous platform. Within the adventure environment, learners are taught basic programming fundamentals like algorithms, pattern identification, iterations, sequence, looping structures and functions. The tool is accessible via a web browser and learners of different age groups can equally use the platform.

Figure 2.7 Box Island Game Tool for Programming

2.7 Activity-based Teaching and Learning

2.7.1 Activity-based Teaching and Learning: What is it?

In the studies of Anwar (2019), Singal et al. (2018), Quin (2012), Margaryan, Collis and Cooke (2003) and Celik (2018) the term activity-based learning (ABL) approach is an established learning approach of actively engaging students in a classroom or outside the classroom to participate in the learning process through discussions, group or individual reflections, case studies, practical exercises, problem-based learning, etc. The necessity that learning be centered on completing hands-on experiments and activities is one of its main foundations.

Historically, activity-based learning began in 1944, during World War II, when David Horsburgh, an inventive thinker and dynamic leader from the United Kingdom, arrived to India and decided to stay. He devised a curriculum that included music, carpentry, sewing, masonry, and gardening in addition to the traditional school disciplines of English, Mathematics, Hindu, and Telugu. These pedagogical resources were meticulously designed, with sketches and illustrations and a touch of fun thrown in for good measure (Wikipedia, 2021).

The concept of activity-based learning stems from the belief that children should be active learners rather than passive consumers of knowledge. When a kid is given the freedom to explore on their own and is given the best possible learning environment, learning becomes more enjoyable and long-lasting. The term activity-based learning is alternatively referenced as active learning (Hazzan, Ragonis and Lipidot, 2020; Quin, 2012). This teaching approach employs different activities before the class, within the classroom and beyond the classroom to engage students practically.

Correspondingly, Silberman (1996, p.2) concluded that in ABL "*Above all, students need to 'do it' figure things out by themselves, come up with examples, try out skills, and do assignments that depend on the knowledge they already have or must acquire".* This approach relates practically to the activity-based approach employed in teaching computer programming.

In fact, using different activity-based approaches consequently enables the lecturers to meet individual learning needs of teaching programming "concepts" among students (Hazzan, Lapidot and Ragonis, 2014). Among the activity-based approaches are quizzes, problem-solving, case studies, educational games, forum discussion, peer reviews, group activities (Anwar, 2019; Singal et al. 2018; Quin, 2012). This approach aids to avert issues in the traditional or conventional method of teaching programming where teachers are seen as being active and just pouring knowledge to students who are mostly seen as passive in the learning journey (Margaryan, Collis and Cooke, 2003; Overmars, 2003; Ali, 2005; Harris, Mishra and Koehler, 2008; Freeman et al., 2013; Byrne, Fisher and Tangney, 2016).

2.7.2 Philosophical Assumptions of an Activity-based Learning Approach.

Learning philosophies as averred in Bigge and Shermis (1999, p.3) and referenced in Abbotts et al., (2008) affirmed that a learning philosophy is: "…*a systematically integrated outlook in regard to the nature of the process whereby people relate to their environments in such a way as to enhance their ability to use both themselves and their environments in a most effective way*".

Studies have proven that there are several learning theories adopted by higher education that build or improve teaching and learning in HEIs. Known among these are the behavioural theory by Skinner (1968) and Cognitive theory by Piaget (1983). The behavioural theory states that "*knowledge exists independently and outside of people*". In other words, a very small amount of knowledge is conveyed to the student whereas learning is achieved when there is a relationship between the learner's motivations and responses. On the contrary, the cognitive theory emphasizes that learning occurs when there is an interaction in the learner's psychological learning environment. Here, the learner can interact with peers and instructors through the acquisition of knowledge. The learner can also learn new things or may ignore old acquired knowledge.

Also, the learning theory as posited by Abbotts et al., (2008) indicates that there is a need to involve a student and a teacher to establish learning. Because of this, a practically orientated approach to enhance teaching and learning to guide a learning process is a good practice to build a learning experience (Bruner, 1961).

Bruner (1961) asserted that the constructivist theory of teaching and learning enhances the learning process and thus promotes an effective students' engagement. Bruner emphasized that the constructivist theory of teaching and learning aids:

1. Tendencies and interest for learning.
2. Teaching and learning process to be delivered in such a way that the student can easily grasp the delivery of the content taught.
3. An effective and dynamic approach to present learning materials sequentially.
4. An effective delivery pace that encourages learners or teachers to give constructive feedback.

Creswell and Poth (2018) affirmed that adopting the constructivist approach enables the researcher to choose a wide variety of processes in the production of knowledge. The constructivist learning theory establishes learning based on the learner's observation and experiences. This approach aids the learner to construct knowledge based on their own understanding. Given this, educational authorities and academics need to design alternative approaches and theories for teaching and learning (Kundi and Nawaz, 2010) in HEIs.

Moreover, the paradigm shift in universities and colleges does not only refer to the shift from the traditional system of educational pedagogies, teaching and learning processes, and management, but it is defined by the quality changes within the learning environments for instruction, administration and learning purposes as well (Moorthy and Arulsamy, 2014). The learning outcomes were carefully analysed by the instructors and consequently, the researcher provided a suggested activity-based approach that aids the teaching and learning of programming. Instructors were to restrain from using any specific activity-based pedagogic approach in delivery, however suggested activity-based approaches such as quizzes, case studies, problem-solving, group discussions, presentations, use of Technology Enhanced Learning (TEL) tools like Socrative, Padlets and Screen Cast O-matic were highly recommended for teaching and learning.

Moreover, each activity-based approach employed comes with its own responsibilities for either the student or the instructor. In most cases, it was suggested that an instructor must do the activities while the students follow the instructions from the teacher as demonstrated in Table 6.3.

2.7.3 Preparing to Teach Using an Activity-based Learning Approach.

Preparing to teach using activity-based learning requires an adequate preparation before, during and after teaching (Celik, 2018; Kosterelioglu and Yapici, 2016; Quin, 2012). Below is a sample demonstration of how the activity-based teaching and learning approach is planned and conducted in teaching students in HEIs. The sample activity-based lesson plan and delivery was developed by the researcher's own professional practice as a lecturer and instructional technologist in a HEI in Ghana.

Sample Activity-based Lesson Plan / Session Planning

Module name: Decision Support System * **Date:** 5/10/2019. ***Level**: Master's Level

Topic: Decision Making and Computerized Support

Description of activity-based approach used: The session consists of lectures using classical lecture style with power point presentation, group discussions, quizzes, problem-solving and videos. This lecture provides students to the introduction of Decision Support System, the characteristics Decision making, Problem solving technique.

Define the Student Profile:

The total number of students in this class are 35 students based on the information derived from the Learning Management System. None of the students suffers any disability, they all possess a laptop. There are no cultural differences between the students. 80% of the students had their first degree in Information technology, however, the remaining 20% have less knowledge in the field of IT. This presupposes some probable difficulties in understanding key technical terms might be difficult among the 20% group with little IT background. Hence

the need to go extra-mile by asking constant feedback and making sure he is at the same level of information with the class.

Learning Outcomes of the session:

By the end of this session, the students will be able to:

a. Understand the conceptual foundations of decision making
b. Understand decision support systems.
c. Recognize different types of decision support systems used in the workplace.
d. Determine which type of decision support system is applicable in specific situations.
e. Understand today's turbulent business environment and describe how organizations survive and even excel in such an environment (solving problems and exploiting opportunities)

A rationale for your choice of method of delivery: (i.e. theoretical underpinning)

The teaching method applied for this topic is the activity-based approach. The rationale is for the students to understand the topic as it is applied in the industry with hands-on practices. Group presentations, quizzes and video presentations will be used. The video presentation will be paused and discussion on the video will be made. Students will be grouped to submit a brief report on the video. The next video will be made available for the students to watch before coming to class the next class.

Timings of the session:

*(This include what you and what the students will be doing at each stage of delivery, a*mend as needed*)*

Time	*Duration*	*Learning content / Activity*	*Tutor / Student What tutor / students are doing*	*Equipment / Resources or Slide number*
08h00	*10 min*	*Welcome and introducing learning outcomes*	*Tutor active / students sign in*	*Oral Presentation*

Time	*Duration*	*Learning content / Activity*	*Tutor / Student What tutor / students are doing*	*Equipment / Resources or Slide number*
8h10	*45 min*	*conceptual foundations of decision making*	*Tutor active / students asking question*	*Powerpoint slide*
8h55	*15 min*	*Discussion on content been taught*	*Watching video / Tutor active / students active*	*Powerpoint slide / Video player system/ Oral conversation*
9h10	*30 min*	*Solving of questions*	*Tutor active / students active*	*Quizzes, Oral and writing on board*
9h40	*35 min*	*Second Delivery Understands decision support systems*	*Tutor active / students active*	*Powerpoint slide and Group discussions*
10h15	*10 min*	*Discussion on content been taught*	*Watching video/ Tutor active / students active*	*Video player system / Oral conversation*
10h25	*25 min*	*Recognize different types of decision support systems used in the workplace*	*Tutor active / students active*	*Case Study, Oral and writing on board and quiz via a TEL tool*
10h50	*10 min*	*Conclusion and Summary*	*Tutor active / students listening*	*Group conversation*

Feedback from the activity-based session

Methods used to collect student feedback:

1) *Google docs*
2) *Socrative for asking multiple views of how the students understand decision as applied in the working filed*

3) *Oral feedback and observation*
4) *Oral questions to check the level understanding*
5) *Homework on the sessions to measure the student's level of understanding*

Analysis of student feedback received on the activity-based approach used

Analysis from the Google doc and Socrative reports on the activities are displayed to students through graphs. Most students are happy with the course delivery using the activity-based approach. They found the lecture very friendly and understandable. Student expressed the desire to even go further by relating the concept decision support system applied in their related work places.

Others also expressed the desire to watch video on DSS applied in their field of work.

Students have then been given assignment to collect some few data concerning the types of DSS apply in their working environment and also to read related literature on DSS.

Personal reflection following delivery

The delivery was good and well conducted.

Given the fact that this course relates to an emerging and fast changing technology, the newest version of the book used should be sought so as to inform the students on the most contemporary issues related to the course.

The engagement of student in solving question together and also asking more question for understanding were positively appreciated

To improve on the delivery and make it more exciting to students, I would like to involve a 10 to 15 minutes video delivery on the next topic

The questions solved in class helped to test about 50% of the set objectives, However, the rest will be assessed through their homework.

2.7.4 Activity-based Learning in Higher Education

The digitalisation of technology has brought about new dimensions for teaching and learning. The study of Chatti et al. (2010) affirmed that learning is individually based, socially affected, shared, found everywhere, flexible, not static and constitutes many different components. Chatti et al. (2010) asserted that "*a fundamental shift is needed towards a more personalised, social, open, dynamic, emergent and knowledge-pull model for learning, as opposed to the one-size-fits-all,* *centralised, static, top-down and knowledge-push models of traditional learning solutions*." (Chatti et al. 2010, p.67).

Merrill et al. (2003), asserted five principles that explain an effective teaching and learning environment. The principles indicate that effective learning is fostered once:

1. Learners are involved in solving a practical problem starting from basic problems to a complex problem.
2. Prior knowledge is initiated as the basis for a piece of new knowledge.
3. The learner acquires new knowledge besides what was taught.
4. The learner applies new knowledge to solve real-world problems.
5. The learner integrates new knowledge to the real world.

The study of Margaryan, Collis and Cooke (2004) established a theoretical framework from the five principles of learning (Merrill et al. 2003) in an activity-based blended learning environment at Shell Open University. Margaryan, Collis and Cooke (2004) established that using technology enhances the balance for interactivity between learners and instructors.

The first learning principle of Merrill et al. (2003) affirms that learning is effective when the learner is involved in the practical problem solving of a real-world situation. This principle is relevant to the field of computer programming instruction where the learners learn how to solve real-world problems from problem analysis to implementation phases of a software. Generally, students find it difficult to program in the early stages of undergraduate computer science programmes (Adu-Manu, Arthur and Amoako, 2013). As a result of this, adhering to the first principle of learning by Merrill et al. (2003), will aid students to perform practical problem-solving techniques within a blended learning environment to accelerate their technical know-how of programming to solve real-world problems.

As a result of the assertions of Merrill et al. (2003), educational institutions need to revolutionize the pedagogic approaches as to how computer science degree programmes are being taught in HEIs. There is a need for HEIs administrators and academics to adopt a new technology-oriented teaching approach to support active teaching and learning (Gros, 2016).

2.7.5 The Relevance of Activity-based Learning (ABL) in Higher Education

Teaching and learning in a traditional face-to-face delivery in HEIs is known to be predominately a learning process where the learner is expected to passively absorb what the teacher instructs the learner to do and to produce back exactly as the teacher instructed (Margaryan, Collis and Cooke, 2004).

Also, Cohen, (1990) asserts that teaching and learning requires a thinking process but what is seen is that "*the thinking required while attending class [traditionally has been] low level comprehension that goes from the ear to the writing hand and leaves the mind untouched*" (Dodge, 1998).

This conservative "chew-pour-pass-forget" approach of teaching has been critically examined by researchers and has faced strong criticisms (Margaryan, Collis and Cooke, 2004; Overmars, 2004; Ali, 2005; Harris, Mishra and Koehler, 2008; Freeman et al., 2014; Byrne, Fisher and Tangney, 2016). Upon the assertions by Margaryan, Collis and Cooke (2004), Freeman et al. (2014), Byrne, Fisher and Tangney, 2016 and Merrill et al. (2003), the researcher feels that the learning process should be more engaging and should support individual development.

Hence, the discovery of the activity-based learning approach has been adopted by most higher education institutions in the domain of engineering, computer science, engineering, social science, mathematics, physics, operations management, etc. (Kanet and Barut, 2003).

Also, Margaryan, Collis and Cooke (2004, p.2) assert that the ABL approach "*provides a way to integrate learning within students' knowledge, and, by exposing them to a variety of activities*". As a result of the high engagement of the learner, it aids both the instructor and the learner to gain a high level of interactivity thereby motivating and coaching the learners rather than just producing content for them. This approach of teaching is very essential to be announced to the students at the very first and introductory session of the course in the class.

Consequently, the researcher agrees with Confucius (551 BC - 479 BC) who once said: *"tell me, and I will forget, show me, and I may remember, involve me, and I will understand"*. This assertion fits perfectively with how computer programming should be taught in HEIs.

Furthermore, the activity-based learning approach is based on the cognitive learning theory and constructivism theory of learning (Dewey, 1938). This learning approach aids the learner to construct knowledge from previous knowledge to more recent knowledge. This learning approach also aids the learner to be actively engaged through the acquisition of the knowledge, assimilation of the knowledge and utilization of the knowledge (Nel 2017; Kanuka and Garrison, 2004; Wulf, 2005; Suhendi, 2018). The teacher or the lecturer is seen as the facilitator throughout the learning process and guides the students through problem-solving procedures and activities (Ali, 2005; Stößlein, 2009). Given this, the ABL approach is seen to be a promising approach to enhance learning experience in higher education institutions.

2.8 Teaching and Learning within a Blended Learning Environment

2.8.1 Blended Learning: What is it?

Blended learning (BL) is the learning approach that integrates both electronic learning and the traditional learning environment. This is also known as the hybrid learning approach (Kintu, Zhu and Kagambe, 2017; Dhawan, 2020; Blanco et al., 2020). The BL approach has come to stay since most HEIs have started an online mode of teaching and learning to make education accessible to everybody, anytime, and anywhere in the world (Dhawan, 2020; Blanco et al., 2020). The delivery mode spans through the use of both synchronous and asynchronous delivery (Chung, Subramaniam and Dass, 2020). With the advent of the deadly Covid-19 pandemic where physical and face-to-face meetings are limited, most HEIs turned out to embrace the use of the BL approach to enhance continuity of teaching (Dhawan, 2020; Blanco et al., 2020). The blended learning environment focuses on the seamless integration of some components of traditional face-to-face teaching learning and an online-based learning experience. With the blended learning environment, students can prepare for lectures by interacting with the online learning resources which in effect reduces lecture time and increases active engagement in the classroom (Chung, Subramaniam and Dass, 2020; Nouri, 2016; Voronina *et al.*, 2017; Alkhatib, 2018).

The blended learning method includes the development of an interactive module designed for each class with short videos designed to review the learning materials, pre-recorded video or audios and links to related sources for learning needs before the class period. Students can submit assignments online, contribute to forum discussions, do collaborative learning (Cripps, Jacobs, and MacCallum, 2020), chat with the lecturer, etc. online. Blended learning adoption requires the student to do personal brainstorming of some key questions, lecture theory, and hands-on exercises, which are communicated to the student using an interactive presentation tool designed to engage the student in lecture activities (Alkhatib, 2018).

2.8.2 Theoretical Rationales of Blended Learning.

The theoretical rationale underpinning the adoption of the blended learning approach as described in the study of Huang, Ma and Zhang (2008); Garrison and Vaughan (2008, p.8) Chung, Subramaniam and Dass, (2020) indicated the following reasons for blended learning usage in HEIs.

Firstly, teaching in larger classes requires the need to develop a blended learning curriculum to accommodate a lot of the students where class size and distance is not be a barrier. According to Huang, Ma and Zhang (2008) the blended learning approach was employed in China as a result of a significant increase in student enrolment. This approach has aided HEIs with greater student enrolment to blend the traditional classroom and online teaching and learning.

Secondly, the blended learning approach is aimed at engaging students outside the confinements of the physical classroom. The study of Huang, Ma and Zhang (2008) asserted that a blended learning curriculum design enables students and lecturers or facilitators to hold discussions online, share ideas among students, perform quizzes at home, interact with teachers and gain access to electronic resources, etc. (Garrison and Vaughan, 2008, p.8)

Thirdly, the blended learning approach is adopted to improve lecturers' and academics' professional knowledge and skills. This is an approach to build employees' skills to enhance efficiency and effectiveness in their profession and consequently meet recent competitiveness in HEIs (Huang, Ma and Zhang, 2008).

In a nutshell, the general goal of the blended learning approach is to enhance learning gains or outcomes and provide cost-effective delivery of the programme as suggested by Garrison and Vaughan, (2008, p.8), Singh and Reed (2001), Huang, Ma and Zhang (2008). Drawing from the three theoretical points of view on blended learning, Merrill (2002) assert that learning is also achieved when the learner is actively engaged in solving a real-world problem. In light of this, the learner gains knowledge as a foundational grounding to acquire new knowledge. This is demonstrated and applied by the learner and diffused to the learner's learning environment or domain. As a result of this, it can be ascertained that blended learning enhances effective instruction of teaching and learning since the student is given a real-case study or problem to solve and also given the right instructions to solve the problem seamlessly via technology.

2.8.3 Blended Learning in HEIs.

The blended learning is a relatively new teaching approach that has been successfully adopted by most HEIs globally especially in the midst of the COVID pandemic (Dhawan, 2020; Chung, Subramaniam and Dass, 2020; Adnan and Anwar, 2020; Kintu, Zhu and Kagambe, 2017; Voronina et al., 2017). This teaching approach integrates the traditional method of teaching and learning with internet-based or online teaching and learning portals (Kintu, Zhu and Kagambe, 2017; Voronina et al., 2017). Blended Learning (BL) in the past three decades gained significant implementation among HEIs because of its ability to employ custom-made learning when developing or designing a course to meet the needs of students for both face-to-face and online delivery (Dhawan, 2020; Chung, Subramaniam and Dass, 2020). To effectively employ blended learning in HEIs, there is the need to have an institutional drive for the teaching and learning within the blended learning environment. In most cases, BL courses are delivered within the scope of an institutional policy (Hargreaves, 2006; Chung, Subramaniam and Dass, 2020).

In spite of the rapid growth of blended learning in HEIs, it does not always guarantee or translate its acceptance into effective teaching and learning within the field of computer programming instructions (Alammary, 2019). Hence, the introduction of the activity-based approach to enhance teaching and learning within the BL environment which this study focuses on. The study centres on how an activity-based approach could be used in teaching computer programming courses within a blended learning environment of HEIs. The BL teaching and learning approach provides the platform for students to watch video resources, download and read lecture materials online, submit assessments online as well as attend a face-to-face lecture. Instructors collaborate with students online; students and instructors can contribute to forum discussions and equally students can meet the lecturer or instructor face-to-face for guidance and instructions. According to Voronina *et al.* (2017), the blended learning approach is a more versatile and interactive learning style intended to yield transformation in learning.

In spite of the relevance of blended learning in HEIs, its implementation requires a rigorous process for institutional adoption, policy implementation and practice (Margaryan, Collis and Cooke, 2004; An, 2013; Myrup, 2017; Chung, Subramaniam and Dass, 2020; Adnan and Anwar, 2020; Blanco et al., 2020). From a personal point of view as the Head for Centre for Online Learning and Teaching at Ghana Communication Technology University, adopting a blended learning approach in HEIs is a very tough decision, which requires dedicated service, financial support and professional orientation for both students and faculty. This assertion conforms to the study of Kintu, Zhu and Kagambe, (2017) that adoption of blended learning using face-to-face and online teaching and learning acceptance is innovative but its development, especially in developing countries, faces huge challenges for its implementation. Hofmann, (2014), asserted that technological failure or difficulties on the part of users usually result in users abandoning the learning process and consequently a failure in the technology adopted for the learning process. Also, individual background and perception affect the adoption of this pedagogical approach to learning (Kintu and Zhu, 2016).

Although, the blended learning approach is good based on literature and practice, there are other challenges it poses to learners with different learning styles, different comprehension levels and technology orientations. An attempt to address these key challenges is the focus of this study by using an activity-based learning approach for the instruction of computer programming courses with a blended learning environment.

2.8.4 The Realities between the Traditional Method of Teaching Vs. Blended Learning Approach

The adoption of blended learning in most higher education institutions has become a paradigm that most university administrators desire to integrate with teaching and learning. In spite of this paradigm shift from the traditional method of teaching, some educators still argue that the traditional method of teaching seems best for them since it has stood the test of time. The study of Nazarenko (2015) affirms that blended learning is growing very fast in higher education worldwide as a result of its technology-driven approaches to augment the traditional method of learning. Nazarenko argues that the blended learning approach did not eradicate the traditional face-to-face method of teaching but rather it has enhanced the delivery of the traditional method using technology to shape the pedagogical experiences.

Several studies have been conducted to affirm the effects of using new technologies in the teaching and learning process with emphasis on how technology enhances the facilitation (Allen and Seaman, 2011; Graham, 2006). Fadde and Phu (2014) indicated that technology integration in facilitation is very good, however, there are several challenges and limitations for employing technology or ICT in teaching and learning. Fade and Phu (2014) asserted that lack of technical and instructional support, face-to-face teaching and learning does not translate directly to online or virtual learning, lack of instructors' motivation to adapt to change, etc. are some of the challenges for employing technology into teaching and learning.

Given these, the realities lie on how the implementation of educational technology approaches can influence learning within the traditional method of teaching. In other words, technology integration with the traditional method is needed to enhance learning seamlessly. There is no doubt as to why the term blended learning is increasingly used among most higher education institutions in the UK and in other countries (Chew, Jones and Turner, 2008b).

According to QAA (2018) it was estimated as of 2003 that 86% of HEIs in the United Kingdom employed blended learning with at least one online learning platform to enhance the traditional face-to-face approach. The percentage marginally increased after two years to 95% of higher education institutions in the UK. As it stands now, it is noted that essentially all universities in the UK have employed a virtual platform to support teaching and learning within the traditional method of teaching (QAA, 2018). This paradigm shift is assumed to support the claims that learners can achieve more and perform better than the traditional method (Kwak, Menezes and Sherwood, 2013).

Literature comparing the traditional method of teaching with the blended learning approach has shed light on how both synchronous and asynchronous delivery can also be used to integrate blended learning delivery to foster learning gains in HEIs using blended learning environment. Table 2.1 demonstrates the difference between face-to-face teaching and learning online or virtual learning.

Table 2.1 Framework distinguishing face-to-face delivery and virtual delivery (Chew, Jones and Turner, 2008)

Face-to-face (Live)	Technology (Virtual)		
	Synchronous (In community)	Asynchronous (In community)	Self-Paced Asynchronous
▪ Instructor-led classroom (lectures) ▪ Tutorials ▪ Hands-on lab ▪ Workshops ▪ Seminars/ Conferences ▪ Coaching / mentoring ▪ Field works / Site Visits ▪ Work-place learning / Placements ▪ 1-to-1 consultation ▪ Examinations	▪ Virtual Classroom / Online Lecture ▪ Online chat / Instant Messaging ▪ E-Conference ▪ Online assessment ▪ Interactive Whiteboard	▪ Discussion board / e-Forum ▪ Announcement / Bulletin Board ▪ Offline message in online chat ▪ Online assessment such as Turnitin ▪ Emails ▪ Search engine ▪ User groups / News groups ▪ Polling and questionnaire or webforms ▪ Blog ▪ Wiki	▪ Online Learning Materials ▪ Online Tutorials ▪ Online self-assessment such as QMP ▪ Podcasts ▪ DVD/CD
	▪ VLE or PLE that consist of more than one element of the above.		
	▪ Online video and photos sharing such as youtube.com, video.google.co.uk and flickr.com.		
	▪ Social Networking such as Myspace, Friendster, Facebook and Ning.		
	▪ Immersive virtual world such as secondlife.com		
	▪ Proprietary software packages and simulations for difference disciplines such as programming simulator, Matlabs and etc.		
	▪ Other general tools such as PowerPoint, flashcard, Camstudio and etc.		

2.9 Review of Existing Models of Teaching and Learning within Blended Learning Environments of HEIs

It is very important to recognize the various models employed in HEIs that aid in enhancing teaching and learning within the blended learning environment. This is because there are different widely accepted models employed by different HEIs. This affirms why Hanson and Clem (2006, p. 137) indicated that *"It is challenging to find a widely accepted definition of blended learning, and even more difficult to find a core set of literature on blended learning mythologies or framework"*. This is because there is no globally accepted standard or framework on blended learning that fits for all subject areas within HEIs.

Blending a course i.e. using a technology-based learning approach and the traditional face-to-face teaching and learning depends on the context and the model of the institution (Nazarenko, 2015). But the underlying factor is that learning within the blended learning environment should be structured to produce the same or similar results of learning outcomes or aims of different professional practices of instructing in the HEIs using different models (Pavla, Hana and Jan, 2015).

Garrison and Vaughan (2008, p.8) assert that "*Blended learning is at the centre of an evolutionary transformation of teaching and learning in higher education. However, transformational growth can only be sustained with a clear understanding of the nature of the educational process and intended outcomes".*

In this section, four different models of learning within the blended learning environment are discussed. These are:

1. The "E-moderation Model" (Salmon, 2000, 2002, 2003, 2013, p. 5)
2. The "Learning Mix Model" (Kaur and Ahmed, 2006)
3. The "Learning Ecology Model" (Wenger and Ferguson, 2006)
4. The "Blended Learning Continuum Model" (Jones, 2006)

2.9.1 E-moderation Model

The e-moderation model as employed by Gilly Salmon (2002) is noted to be one of the most prominent models employed by most researchers in the domain of e-learning and online education delivery in the UK for the past two decades. Her model brought about a paradigm shift in HEIs in the UK when she introduced a model called "e-tivities and e-moderation". The e-moderation model provided a five-stage process that outlines a step-by-step approach for the delivery of an online learning approach. Salmon (2004b) indicated that the e-moderation model is widely accepted by most academics globally to enhance teaching and learning. This is because the e-moderation model as propounded by Salmon was built based on Abraham Maslow's hierarchical theory of needs (Maslow, 1943). Given this, it will be of good relevance to bring to light how Salmon's framework of e-moderation fits into Maslow's theory of needs.

Maslow's (1943) hierarchical theory of need is known to be one of the most recognized theories when it comes to assessing the needs of humans. It is also an undeniable fact that the model is also applicable in the field of education that underpins the motivations of effective teaching and learning among students and teachers (McLeod, 2007; Chew, Jones and Turner, 2008; Freitas and Leonard, 2011; Millheim, 2012). Figure 2.8 indicates how Maslow's theory fits into higher education.

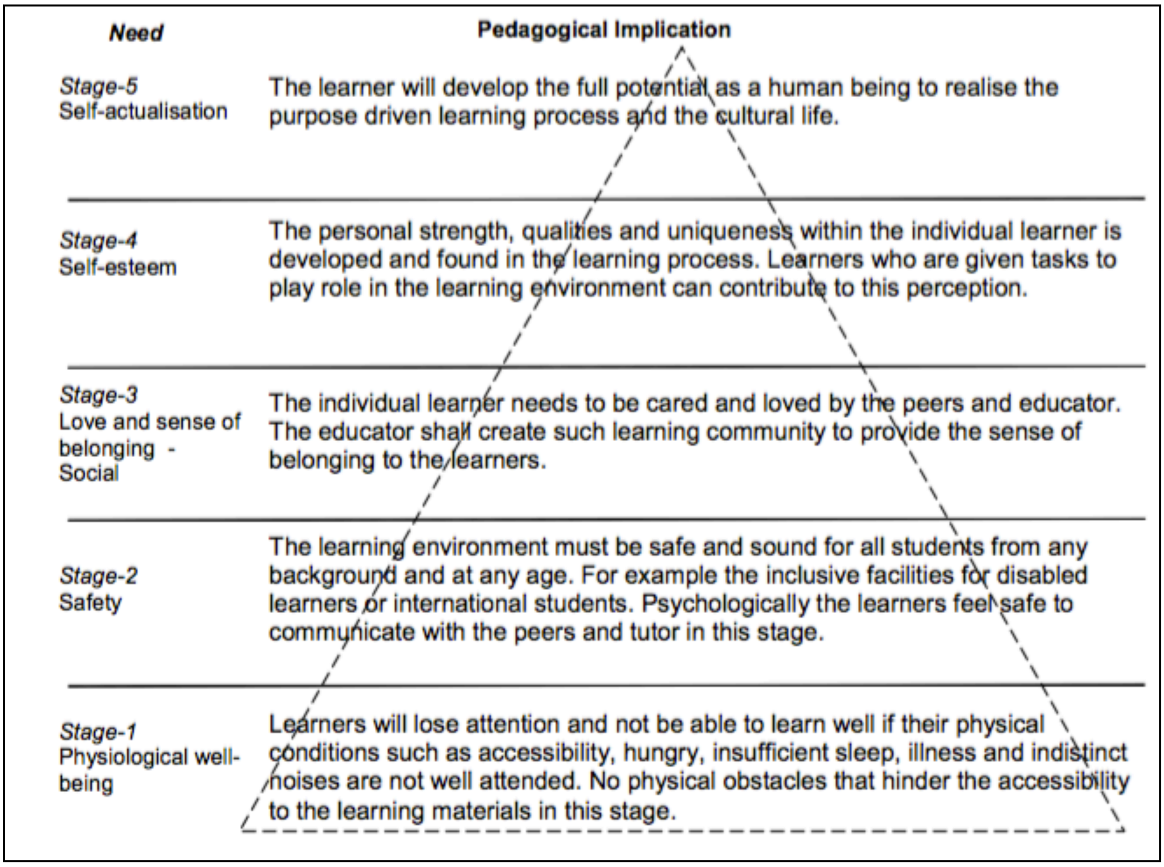


Figure 2.8: Abraham Maslow's Hierarchical Theory of need and Pedagogic Effect (Source: Alexander, 2006 and Chew, Jones and Turner, 2008)

In an attempt to interpret Salmon's model of e-moderation in relation to Maslow's hierarchy of needs, researchers such as McFadzean (2001); McLeod (2007); and Millheim (2012) asserted that self-actualization and learners' personal development are key factors in an individual's educational fulfilment. This conforms to the five-stage process of learning as propounded in Salmon's model of e-moderation as indicated in Figure 2.9.

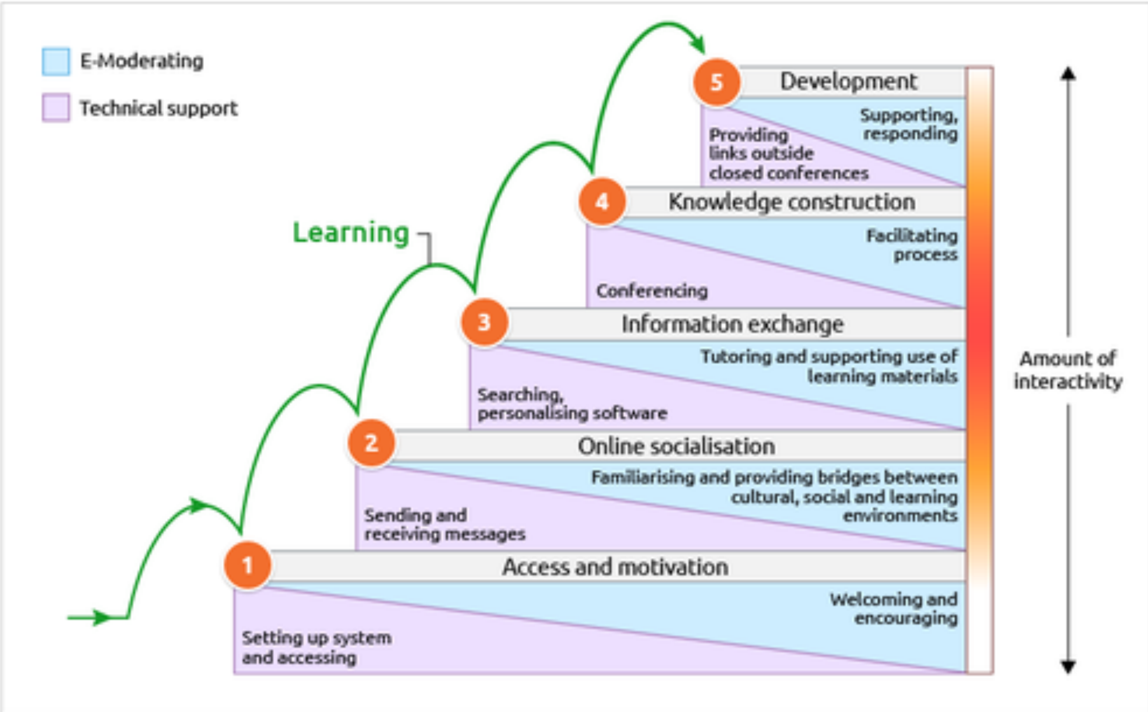


Figure 2.9: Salmon's Model of E-moderation, Salmon (2000, 2001)

The very first stage of Salmon's model talks about granting to the learners a welcoming and a warming encouragement to make learning appealing to the students. The model as widely used by UK Open University (Salmon, 2001), usually uses the first stage of Salmon's model to walk learners through how they can interact on the online learning platform with their new environment and the technologies that will be used to enhance learners to their learning goals. In other words, learning platforms and welcoming motivations are given at the first stage of Salmon's model.

The second stage of Salmon's model describes the need to enhance socialization and familiarization of learners to their new learning environment. At this stage, a robust bridge is built to establish the learner's social and cultural attitudes to their new learning environment devoid of any forms of impediments. This is where learners can send and receive messages among peers and teachers. The driving force at this stage is to enable learners to interact among peers and teachers after reading and accessing online learning resources or materials available to them seamlessly.

The third stage of Salmon's model establishes consciousness of the learners belonging to the learning community. This is where the learners can make a personal search for additional materials or resources that may contribute to the learning journey of the student. The third stage of the model works parallel to that of the fourth stage. At the fourth stage of the model, learners' confidence in the knowledge they have achieved is enhanced. Here, learners start to construct knowledge on their own, contribute and even lead a group discussion and express personal opinions of arguments based on the knowledge acquired.

Finally, the fifth stage of Salmon's model establishes a condition of the learner where the learner now becomes responsible for their learning journey. Individual learners at the last stage become confident, self-reflective, self-motivated and consequently achieve self-actualisation using the e-moderation model.

In spite of the positive claims, Salmon (2004) established on the e-moderation model to have adequately provided quality online learning and teaching in the past two decades, it has also faced critique from researchers. The study of Moule (2007) challenged the model that although it is becoming a prevailing discourse and has been adapted by most institutions employing online learning and teaching and cited by several researchers, the model cannot be the solution when it comes to e-learning and online teaching and learning. Moule (2007) contends that Salmon's model is limited considering the e-learning approach employed in the domain of computing education by neglecting the various theories of learning within a computer-mediated communication. Moule (2007) argues further that, not all e-learning follows learning within a community as Salmon claims. Also, I agree with Lisewski and Joyce (2003) and Moule (2007) that Salmon's model does not support face-to-face teaching and learning within a blended learning environment.

Besides, Chowcat (2005), Lisewski and Joyce (2003) affirmed that Salomon's e-moderation model is very rigid and thus, always difficult to transfer to other learning situations. Furthermore, Hammond (2007) and Chew, Jones and Turner (2008) assert that although the socialization and communication aspects of Salmon's model is good, yet you cannot force learners to make meaning of what others posit in the online discussions. Also, they affirm that based on their personal experience on the model, a learner can just read a post of a colleague within the learning community without actually participating in the discussion or can just post anything because it is required of the learner. In all the e-moderation is an interesting model for online teaching and learning.

2.9.2 The Learning Mix Model

The Learning Mix model is also known to be one of the profound models used within the context of blended learning (Kaur and Ahmed, 2006). This model was first used by the Open University of Malaysia (OUM) that implemented the “Open” learning that was supported by an association of eleven state-owned universities in Malaysia (Chew, Jones and Turner, 2008).

The model employs a blended learning approach that includes the integration of face-to-face learning, online learning and self-managed learning. Figure 2.10 below depicts the blended learning framework adopted by OUM as designed by Kaur and Ahmed (2006). This approach has been widely used by numerous researchers globally. The blended learning approach is noted to be a convenient approach that promotes learning not only in the confinement of the classroom or online but a seamless learning that combines both the face-to-face and online engagement (Chew, Jones and Turner, 2008; Poon, 2014; Nazarenko, 2015; Pavla, Hana and Jan, 2015; Kasame, Pachoen and Manit, 2016; Kharb and Samanta, 2016).

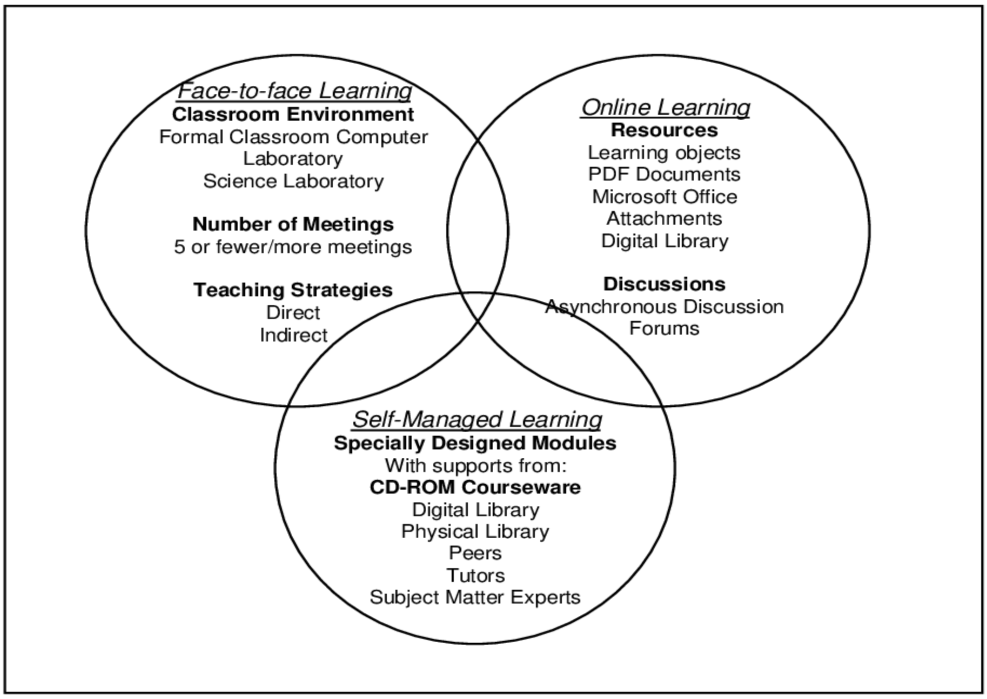


Figure 2.10: The Blended Learning Mix Model (Kaur and Ahmed, 2006)

According to the model by Kaur and Ahmed (2006) on the learning mix model, they asserted that the face-to-face component of their learning provides to the learner, the social interactivities among colleagues and instructors. This is where the learners gain direct interaction with their facilitators or instructors and peers, attend formal classroom lectures, perform activities in the classroom, the computer laboratory, science laboratory, etc. This learning model is a limitation to Salmon's model of e-moderation.

On the other hand, the online learning component augments both the face-to-face interactions and self-managed learning. The activities performed on the online learning platform include the learner gaining access to online learning resources such as PDF lecture notes, video tutorials, contributing to an asynchronous forum discussion, online submission of assignments, etc.

Kaur and Ahmed (2006) continued to explain that the model provides to the learner a self-managed learning approach where additional learning resources are printed for the student or stored on CD-ROM and other digital media and given to the student. As this model was employed by OUM, learners are required to spend at least 2 hours each day to read the learning resources.

The rationale for the blended learning mix model is to aid effective collective learning among learners. This model enables the learners to learn seamlessly, i.e., learning can be achieved anywhere and at any time within the learners' own learning pace and time (OUM, 2019, Okaz, 2015; Olsson and Mozelius, 2016). However, it is noted that self-discipline is very essential to carry the agenda of this model. Since a greater percentage of the learning process depends solely on the learner's self-motivation and self-discipline, it poses a challenge to learners to achieve learning goals, i.e. learning development as Salmon (2001) indicated in the five-stage model is however recognized very little when using the blended learning model.

Although the blended learning mix model is known to be a very good approach to effective teaching and learning in higher education, Okaz, (2015) argues strongly that it rather causes more harm than good. Okaz, (2015) indicates that learners on the blended learning model mostly do not find the classroom engagement motivating enough to attend classes.

Consequently, Okaz (2015) asserts that the learners rather cause a lot of behavioural issues and hence are not interested in participating in the class at all. The rationale behind Okaz's claim is that the increasing presence of technology such as online games, smartphones, social media applications, etc. has distorted learners' attitudes and behaviour and has thus changed the way and manner students even communicate in the classroom and outside the classroom. This is a worrying phenomenon that forces teachers to restructure the learning process to suit the drift of most students.

Moreover, Vonderwell (2003) indicated that some students on the blended learning mix model feel disconnected in the class as a result of the computer-mediated approach of delivery. Likewise, some students as a result of the difference in the economic situation and different levels of Information Technology know-how may not feel they belong in the blended class because they cannot connect to the online platform (Holley and Oliver, 2010). Also, issues of age and gender were seen in the studies of Khechine, Lakhal, Pascot, and Bytha (2014) who asserted that young students can more easily adapt to learning within blended learning than older-aged learners.

2.9.3 Learning Ecology Model

Wenger and Ferguson (2006) assert that the learning ecology model supports a broader and comprehensive perspective of teaching and learning. Figure 2.11 below describes the learning ecology model. The model factors a quadrant representation of teaching and learning. These are teaching, studying, practising and coaching. All these learning paradigms are guided by broad and complete learning processes towards an effective content delivery, learning experience, and practice.

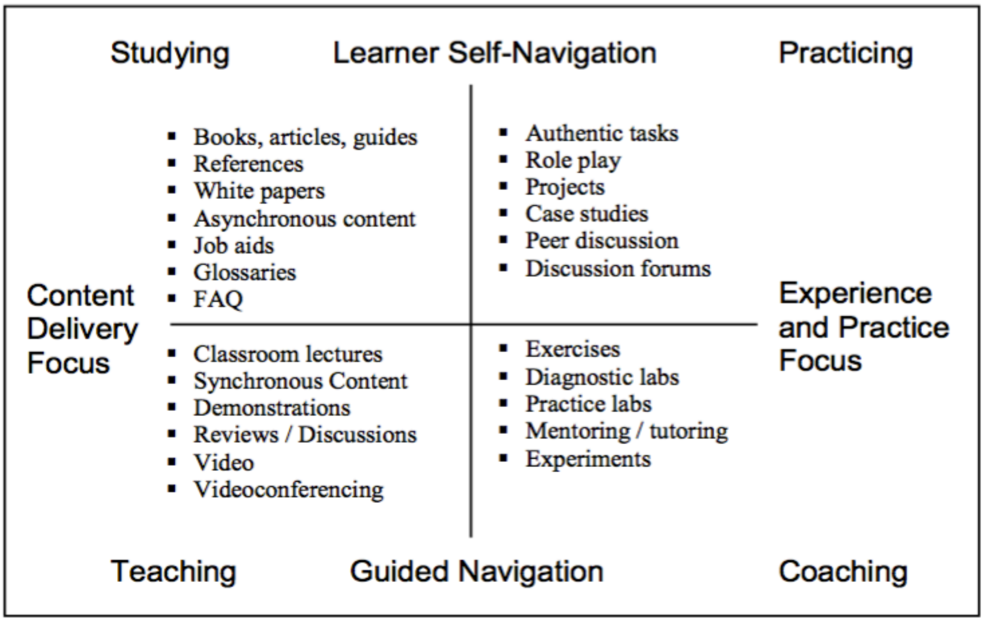


Figure 2.11: Learning Modalities (Wenger and Ferguson, 2006)

The model proposed five identifiable backgrounds in the learning ecology model (Wenger and Ferguson, 2006). These are;

1. Provision of quality of learning.
2. Learner gaining control over the learning experiences.

4. Informal learning vs formal learning.
4. Provision of social content that supports learning.
5. Provision of a cost-effective delivery.

The above five essential indicators were used to construct the framework that provided enablement of teacher-oriented and learner-oriented learning experiences. Also, the framework provides a flexible nature of learning that engages the learner for practical and social contributions within the learning environment. However, there are no measures in place from the model to ascertain that the model provides quality delivery (Chew, Jones and Turner, 2008).

Also, the model has faced critique from Chew, Jones and Turner (2008), arguing that the model does not provide a dynamic method of learning. According to the studies of Siemens (2003); Khan (2011); T. Huan, Shehane and Ali, 2011; Brahimi and Sarirete, (2015); Flores, del-Arco and Silva, (2016) teaching and learning should be more dynamic. It was however noted that the learning ecology model failed to present a more dynamic approach for teaching and learning. For example, a learner can decide to engage in peer discussions, or respond to case studies without necessarily gaining coaching from the instructor or the teacher. The model failed to differentiate between what it means to be self-navigated as against guided navigation and thus remained very confusing. Also, several researchers have introduced the Vygotsky model of learning by providing a comprehensive and overlapping approach of learning which the ecology model failed to adapt (Jaramillo, 1996; Ussher and Gibbes 2002; Gredler and Shields, 2004; Leong and Bodrova, 2007).

2.9.4 Blended Learning Continuum Model

According to the studies of Jones, et al. (2009) the University of Glamorgan (UoG) located in the UK is known to be one of the higher education institutions that has employed blended learning in the whole of the university. In other words, there was an institutional drive to move the whole university to a blended learning environment. Chew et al., (2006) indicated that the entire paradigm shift of the education delivery at UoG which was led by Prof. Norah Jones took the university 3 years to complete. Jones (2006) indicated that the blended learning scale is a well-guided continuum that enhances teaching and learning rather than a stage-like framework for its adoption in the entire university. The scale or continuum used by UoG is indicated in Figure 2.12.

The first indication of the continuum is the “Basic ICT usage” which is followed by “E-enhanced” scale within the blended learning environment. At these two stages the uses of basic ICT tools like PowerPoint presentations, MS-word, projector and a web-based learning platform or virtual learning platform enables students’ and lecturers’ communication. The third indication on the continuum is the “E-focused” point which is much more inclined toward active online learning engagements. At this point students are now able to make submissions of assignments via the online learning platform, contribute to discussion forums, perform online quizzes and tests and gain access to interactive learning resources. Jones, et al. (2009) explain the last stage of the continuum as “E-intensive”. At this phase, the teaching and learning are fully delivered online with the integration of the face-to-face component of learning.

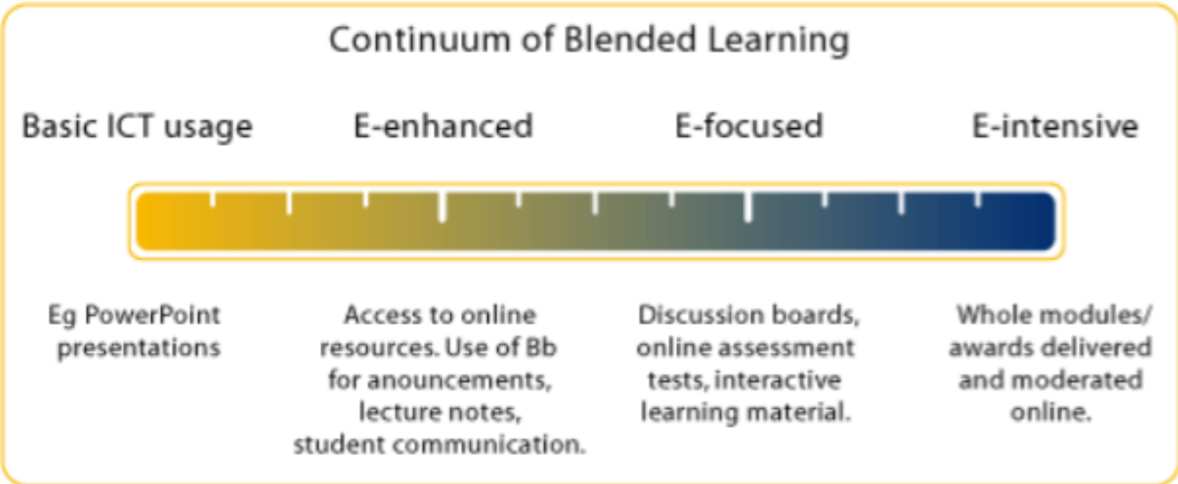


Figure 2.12: Continuum of Blended Learning (Jones, 2006)

Upon critical evaluation of Jones’ continuum model of blended learning in HEIs, it can be deduced that this approach fits better than the blended learning model of ecology by Wenger and Ferguson (2006) and the Blended Learning Framework as adopted by OUM (Kaur and Ahmed, 2006). The reason for this argument is that this model provides more flexible, and practical adoption of technology in teaching and learning. Additionally, the approach enables the facilitators or practitioners to choose which point on the continuum to apply at a particular time to meet the learning needs of the student and the epistemological dimension of the learning disciplines (Jones et al., 2009, Chew et al, 2008).

Additionally, there is a clear similarity that exists between Jones’s model and that of the community of inquiry model by Garrison and Vaughan (2008). The drawing point here is that both frameworks reject the dualistic assumption of opting for traditional face-to-face teaching and learning and online learning. In other words, both frameworks can swing between their choices of what to do within the blended learning environment and thus provide a flexible approach to teaching and learning. Moreover, Jones (2006) affirmed that the very stage on the continuum i.e. “E-intensive” is not necessarily noted as the best approach to blended learning but rather a guide. Hence, it can be changed depending on the subject because of its explicit approach to adapt.

Additionally, other researchers such as Allen et al. (2007) and Chew et al. (2008) confirmed and validated the framework of Jones (2006). Allen researched in the US and the findings provided were very similar to those of Jones. Allen concluded that traditional learning, web-based facilitation, blended or hybrid learning and online learning contributes to effective blended learning. Table 2.2 below summarizes why both Allen et al. and Jones’s approaches are complementing each other.

Table 2.2 Similarities of Allen et al. (2007) and Jones (2006) Continuum Model of blended learning.

Percentage of Delivery %	**Allen et. al (2007) learning Approaches**	**Jones’s, (2006) Continuum model**	**Summary of the Adopted Approaches**
0	Traditional	Basic ICT Usage	Courses are delivered without any online platform. Mainly face-to-face and oral delivery
1-29	Web-based facilitation	E-enhanced	Using online learning platform or web-based technologies to facilitate learning. E.g. Using learning management systems (LMS), Course management system (CMS)
30-79	Blended/ Hybrid learning	E-focused	This is where the conventional face-to-face approach are blended with the online learning platform. i.e. A component of the learning is done in the classroom while other components are done online using an LMS or CMS.
80+	Online	E-intensive	This is fully an online learning engagement where course delivery is mainly conducted online without face-to-face engagement

An introspective study on Jones's model and Allen et al.'s approaches of learning as indicated from Table 2.2 above indicates that the first indication on Jones's continuum is in line with the traditional method of delivery of Allen et al. where no online tool is used in the classroom. Similarly, the E-enhanced indication also maps with the web-based facilitation in Allen et al.'s study. Likewise, the E-focused indication also maps with the blended or hybrid approach and lastly, e-intensive matches with the online learning model of Allen et al. (2007).

However, the indications of the percentage of learning delivery does not make any meaning in Allen et al.'s argument in the sense that, the percentage of learning delivery of an instructor to engage students or learners is a sole prerogative of the instructor. Cross (2006) also supports the same assertion that Allen et al., are too simplistic in quantifying the percentage to be used in any of the learning approaches. It is also noted that Jones's model was very silent on how activity-based learning could be employed within the blended learning environment. Notwithstanding these critiques, the model is seen to be very flexible and can afford diverse variations in the context of blended learning delivery.

Finally, on the continuum model, Chew et al. (2008) affirmed that the model is mostly driven towards technology rather than the pedagogical approaches of teaching and learning. Employing such a framework in the field of computer science education where a lot of engagement is needed by the students to perform hands-on learning in the classroom and online cannot be fully practicable. The pedagogic approaches needed to deliver courses in HEIs were not factored into Jones' continuum model. Furthermore, the continuum model was very silent on the responsibilities of the learner and the instructors as Wenger and Ferguson (2006) employed in the learning modalities model.

In a nutshell, all the above four models have contributed significantly to the development of blended learning adoption in higher education institutions. It was however noted from the existing BL models that the application of the activity-based teaching and learning tailored towards teaching and learning computer programming was not strongly highlighted.

2.10 Technology Enhanced Learning (TEL)

One of the key developments based on pedagogical approaches in teaching computer programming is the adoption of Technology Enhanced Learning (TEL) (Goodyear and Retalis, 2010; Kirkwood and Price, 2016). TEL is a term mainly used in the UK and other parts of Europe. The TEL approach integrates technology with the aid of smart devices (ipads, tablets, notebooks, laptops, etc.) for teaching and learning. The technology enhanced learning tools can be employed within the classroom, before the class, and after the class. The technology enhanced teaching and learning tools are mostly driven by Internet enabled applications. Figure 2.13 below depicts examples of TEL tools and how they are applied for teaching and learning as used by the researcher's own professional practices as the Head of the Centre for Online Learning and Teaching (COLT) at Ghana Communication Technology University (GCTU).

United Kingdom (UK) Universities and the Associations of Information Systems and Computer Science Education, indicate that TEL is "*any online facility or system that directly supports learning and teaching*" (Walker, Voce and Ahmed 2012, p.2). In other words, it is a teaching and learning pedagogy that employs the use of online tools to engage students towards enhancing their understanding of the subject matter.

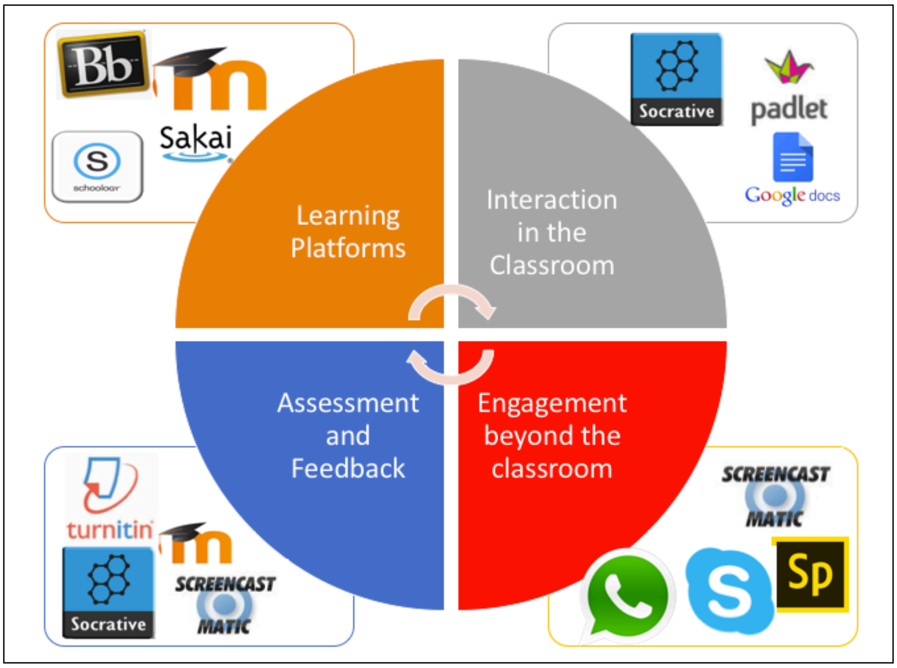


Figure 2.13 Technology Enhanced Learning Model within a Blended Learning Environment, Author (2019)

The application of the above TEL model as developed by the researcher has gained a lot of attention by most lecturers from different disciplines i.e. business, engineering and computing at GCTU in Ghana and other university lecturers who have gained training from the researcher on these TEL tools.

2.10.1 Learning Management System (LMS)

Learning management system refers to software systems purposed for administering, documenting, broadcasting, reporting, pursuing and provisioning of educational content or courses (Davis, Carmean and Wagner, 2009). In Higher Education Institutions (HEIs), learning management systems have become critical to teaching and learning (Matarirano et al., 2021). The rationale for LMS is to promote the delivery of teaching and learning via a technology enhanced learning platform. LMSs as initially introduced in the 1990s have gained popular attention in recent times among HEIs globally (Coates, James and Baldwin, 2005; Dalsgaard, 2006; Vázquez Cano and Gisbert, 2015; Olsson and Mozelius, 2016; Klašnja-Milićević et al., 2017).

The LMSs provide learning platforms, Course Management Systems or Content Management Systems (CMS), e-portals, etc. which constitutes a wide range of courses and pedagogical components to provide seamless teaching and learning via the Internet (Coates, James and Baldwin, 2005; Dalsgaard, 2006). The LMS platform provides activities (e.g. assignments, forum discussions, quizzes, Turnitin submissions, group chatting etc.) and resources that promote teaching and learning. Learning resources accessible to students can either be in text (e.g. PDF, DOCX, DOCS, etc.), multimedia resources (e.g. videos, audio, podcast, etc.), links to Universal Resource Locators (URLs).

The researcher strongly believed that to be able to teach effectively within a blended learning environment, there is the need to firstly acquire a Learning Management System (LMS) (Matarirano et. al, 2021). Examples of LMS that have gained global recognition of usage among HEIs are Moodle, Sakai, Blackboard, Schoology, Desired2Learn (D2L), Google Classroom, etc. A 2019 research study by Market Research and cited by FinancesOnline (2019) affirms that a lot of higher education institutions and companies have embraced LMSs to enhance learning solutions. They estimated that by 2021 the global LMS adoption and the e-learning market will balloon to $15.72 billion based on the previous three years steady growth rate of 5% and $22.4 billion by 2023 respectively. Moreover, the Zion Market Research unequivocally affirmed that the LMS market will increase by $19.05 billion based on a 5-year prediction of a steady 24% Compound Annual Growth Rate (CAGR) (Finances Online, 2019).

Nonetheless, the LMSs do not provide a pedagogic approach that can enable teachers to follow but rather provides a content-based platform where resources and activities can be shared to students. That is, using an LMS does not ensure that teaching and learning is effectively delivered. Hence, the researcher sought the need to augment a technology enhanced platform (i.e., the LMS) to foster the teaching and learning of computer programming in HEIs using an activity-based approach.

2.10.2 Engagement in the Classroom

Another aspect of using TEL in HEIs is the employment of TEL applications that promote students' engagement and interaction in the classroom (Nel, 2017). Examples of the TEL tools that promote interactivities and engagement in the classroom include Socrative, Padlet, Google Doc, etc. Socrative is a "*smart student response system that empowers teachers to engage their classrooms through a series of educational exercises and games via smartphones, laptops, and tablets*" (Socrative, 2019).

Socrative can be used to conduct quizzes, long answers to questions and answers (Q and A), open-ended responses to questions, conducting voting, anonymous surveys, conducting pre-testing and post-testing, etc. Also, in a more recent study of Tirlea, Muir and Elphinstone (2018) they asserted that students professed that Socrative had a positive impact on their learning experiences, engagements, and interactivities. It was also noted in their studies that Socrative improved their learning and retention.

Also, Padlet is used to enhance collaborative learning by sharing ideas of students and teachers, brainstorming, creating online notice boards, sharing resources, etc. The rationale for using these TEL tools in the classroom is to effectively engage the students for teaching and learning (Roffe, 2002; Khan, 2011, Tirlea, Muir and Elphinstone, 2018)

2.10.3 Engagement Beyond the Classroom

The Technology Enhanced Learning approach is also noted for its essential use even beyond the classroom environment to enhance teaching and learning at the comfort of the learners and the teachers. The TEL tools used for this approach are noted for their significant contribution to learners with learning difficulties during the face-to-face lecture in the classroom by the provision of remote support to the student (Nouri, 2016). Some applications that support the teaching and learning delivery beyond the classroom include social networking tools like WhatsApp, Twitter, Zoom, Skype, among others. Students can also be engaged beyond the classroom using Adobe Spark and Screencast O-matic for preparing short videos to engage students before coming to the next face-to-face class.

Consequently, students are able to learn within their own learning pace and time and freeing up classroom hours for individual creativities and active-based learning. This also aids the lecturer or facilitator to gain an opportunity to interact (one-on-one) and assess the learner's learning growth thereby promoting the student to be responsible for his or her own learning experience (Gilboy, Heinerichs, and Pazzaglia, 2015; Betihavas et al., 2015).

2.10.4 Assessment and Feedback in TEL

Assessments and feedback are very essential elements in the delivery in teaching and learning (Boud and Associates, 2010; Carless et al., 2011). This is because the assessment and feedback component of teaching and learning engages learners to be productive in their learning journey. Studies have affirmed that there are no explicit methods of conducting a quality assessment. This implies that assessment activities must conform to best practices, standards, and detailed assessment criteria (Boud and Associates, 2010; Carless et al., 2011; Fluckiger et al., 2010; Gibbs and Simpson, 2004; Gilbert et al., 2011; Miller et al., 2010; Sadler, 2010).

The feedback activities are considered as a progressive activity that a student acts upon within a particular period of time (Boud and Associates, 2010; Carless et al., 2011; Fluckiger et al., 2010; Gibbs and Simpson, 2004; Gilbert et al., 2011; Miller et al., 2010; Sadler, 2010). The feedback activities on the other hand inspire students to reflect on their individual academic journey and otherwise make amends, improve or maintain the learning experiences (Lew, Alwis, and Schmidt, 2010; Nicol, 2008).

Price and Kirkwood (2011) assert that assessment and feedback work together to effectively carryout quality delivery of assessment to students. Examples of TEL tools that support quality assessment and feedback practices include Turnitin, Moodle, Socrative, Screen Cast O-matic, Sakai, etc. Mohanty and Vohra (2006) and Price and Kirkwood (2011) posit that higher education institutions must employ the use of Turnitin to check plagiarism or similarity index of students' assessment submissions. These tools can be used to perform both summative and formative assessments (Price and Kirkwood, 2011). Lecturers can further give instant feedback to the students using text, audio or recorded video to grant feedback to the student(s) via the LMS. Also, examples of assessment tools that support audio-visual feedbacks are Adobe spark, Screen Cast O-matic, Padlet, etc.

Additionally, Price and Kirkwood (2011) assert that issues relating to enhancement still need further investigation by addressing the following questions: What should be enhanced? How should it be enhanced? When should it be enhanced? And who should enhance it? Also, the use of TEL cannot measure whether teaching and learning affects students' performance or not. Therefore, there is still a research gap in the adoption of the TEL approach which needs further scrutiny to enhance effective teaching and learning.

2.11 Smart Learning

The smart educational approach as depicted in the study of Zhu, et al. (2016, p. 15) is to enhance students' or learners' quality of learning in HEIs. Zhu, et al. (2016, p. 15) defined Smart learning as "*…smart education is to improve learners' quality of lifelong learning. It focuses on contextual, personalised and seamless learning to promote learners' emerging intelligence and facilitate their problem-solving ability in smart environments*". According to Sharples et al. (2014) the concept of seamless learning is the application of individual usage of a continuous learning environment where teaching and learning can occur anytime, anywhere, using different technologies or other social situations and where learners can assess learning materials and collaborate with other students. Therefore, the smart learning approach is a student-centred educational paradigm.

This learning approach also confirms the study of Middleton (2015) who pointed out clearly that smart education must be developed to enhance students' learning experiences and student centeredness. The study of Lee (2015) asserted that smart learning constitutes the integration of technology into teaching and learning within formal learning, informal learning, collaborative learning and social learning.

Also, smart learning has become employable in the business working environment. This approach of learning is grounded on the Internet of Things (IoT). The IoT-based approach enhances a seamless amalgamation of different opportunities and tools that support learning. Figure 2.15 depicts an iterative IoT-based approach that supports an accelerated learning without any form of boundary i.e., place, time, etc. To approach the smart learning processes, the following processes are employed:

1. identify the user's learning need,
2. establish the toolkits that support learning,
3. develop a blended learning platform that can support learning seamlessly, and
4. provide both analogue and digital learning tools that support IoT physical devices.

Consequently, the learning objectives and learning contents will be made available to the learner via the online platform and users can seamlessly connect to the learning via the IoT devices.

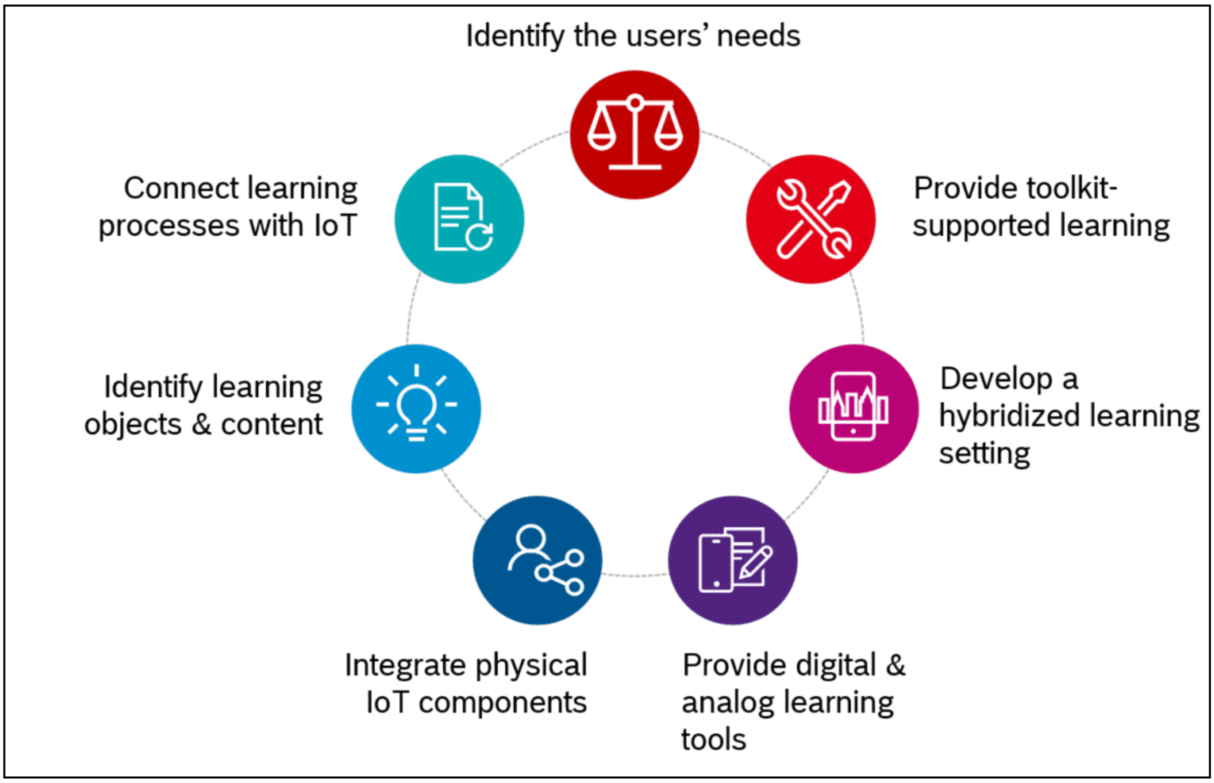


Figure 2.15 Approaching Smart Learning Environments (Bosch Software Inc., 2019)

It can be deduced that although the smart learning approach is seen to be an innovative approach for teaching and learning, there is still the need to employ other pedagogic approaches to fully operationalize the smart learning approach in HEIs. Burbules (2012), indicated that for teaching and learning to be effectively practised ubiquitously, it is required of the teacher and learners to have more time and space to build the learning experience and thus the need for an adaptable framework to employ. Also, different economic backgrounds sometimes hinder the effective implementation of the smart learning approach considering the devices and applications used to enhance learning.

2.12 Conclusion

The chapter has shed light on the prevailing teaching and learning strategies employed in teaching computer programming in HEIs. Consequently, the reviewed literature has shed light on the teaching strategies of programming such as the activity-based approach, problem-based approach, puzzled-based approach, game-based approach and pre-recorded approach. Also, the review shed light on the critical challenges faced by students in computer programming courses which needs attention and resolution, hence, the need for this research to overcome the prevailing challenges in computer programming instruction by developing a framewok for an activity-based computer programming instruction within a blended learning environment.

Furthermore, the reviewed literature has shed light on the fact that there are research gaps in the adoption and practice of an activity-based learning approach within the blended learning environment. Also, the novel approaches of activity-based learning, blended learning and the use of technology-enhanced teaching and learning in higher education have been underscored as a paradigm shift in HEIs. It was deduced from the literature that teaching computer programming still follows the traditional method of instruction and has affected the retention rate and performance of most students in HEIs of Ghana.

The various blended learning models used in HEIs have also been noted in the literature. However, the literature did not shed much light on the application of the activity-based learning approach in the blended learning environment, hence the need for this research to enhance effective teaching and learning in HEIs in computer programming courses in HEIs.

CHAPTER 3

THEORETICAL AND CONCEPTUAL FRAMEWORK

3.1 Introduction

This chapter discusses the conceptual framework that underpins the development of an activity-based instructional approach for teaching computer programming within a blended learning environment. The conceptual framework for this research was developed based on an existing framework derived from literature in an attempt to answer the research question 4 of the study; that is, what new framework could be used to enhance an activity-based learning approach in a blended learning environment for instructing computer programming?

The proposed conceptual framework underpinning this research outlines the key components inferred from the literature review, expert opinions and personal experience of the researcher. The framework was validated by experts and has been presented in three different PhD symposia at UNISA School of Computing and the South African Institute of Computer Scientist and Information Technologist Conference (SAICSIT), 2019 (see Appendix L). A rigorous approach was taken to develop the activity-based teaching and learning framework for computer programming instruction within a blended learning environment. This chapter thus, presents the theoretical and conceptual framework that underpins the activity-based approach of teaching computer programming within a blended learning environment.

3.2 Learning Theories

Learning theories define a framework which serves as a root to the thoughtful acceptance of how people learn, describe, explain, perform analysis towards the establishment of making the right decisions about the design, progress and delivery of learning (Goel, 2017). Learning theories have diverse forms and approaches. Examples of these learning theories are the cognitive theory of learning, behaviourism theory of learning, constructivism theory of learning and social constructive theory of learning.

Goel (2017) alluded that "*Learning theories offer frameworks that help understand how information is used, how knowledge is created and how learning takes place. Learning* *designers can apply these frameworks according to different learning and learner needs and make more informed decisions about choosing the right instructional practices".* It is noted from the literature that there is no one perfect learning approach. This is because, there are different learning environments, contexts, scopes, styles, etc. Hence, there is no "one-size-fits-all" and best approach for all cases of learning. However, any of the theories can be applied in the teaching and learning of computer programming (Malmi et al., 2019). In this study, the researcher presents different learning theories used for teaching and learning in higher education that have influence on the teaching and learning of computer programming. The researcher beliefs that the application of the constructivism theory, behaviourism theory, cognitive learning theory and the social constructivism theory influences the teaching and learning of computer programming in HEIs.

3.2.1 Constructivism Theory

Constructivism learning theory is defined as the active construction of new knowledge based on a learner's prior experience (Suhendi and Purwarno, 2018; Alzaghoul, 2012). Ben-Ari (1998) posits that the constructivism theory assumes that students can better construct knowledge through active engagement and not just passively receiving information from the instructor or teacher. Under this theory, the role of the teacher is not only to observe the students and examine them but in addition to that, actively engage the student through asking of relevant questions to challenge the students to reason as they work to complete activities in class. The learner plays an active role in the learning process according to this theory through the building of understanding and making meaning of the information provided (Woolfolk, 2007). The belief of the constructivist is that reality is not a single version but rather made up of a mass of realities that can be found in each learner. Thus Phillips et al., (2008) argue that learning is contingent on the ability of the learner to undertake proper analysis, synthesis and evaluation of information to arrive at meaning and finally personalize the knowledge. It therefore implies that the student or learner can come to authentic learning through active engagement in the learning process and the discovery to transform the learner to understand complex concepts and apply it in the learner's own context (Slavin, 2003).

This approach to learning in which learners apply the information in developing practical solutions is important for the learning and developing of practical skills such as computer programming (Suhendi and Purwarno, 2018). This is because effective teaching and learning of skill-based programs such as the learning of programming is based on constructivism in which students are expected to be engaged in the learning process in addition to actively participating in practical, critical thinking and problem-solving learning activities (Wulf, 2005; Price, McFadden and Marsh II, 2001; Kundi, and Nawaz, 2010). Learners therefore actively construct their own knowledge as they test ideas that are based on their earlier knowledge and experience, apply these to a new situation, and integrate the new knowledge gained with pre-existing intellectual constructs (Ben-Ari, 1998). Hence, the researcher applied the constructivist learning theory for the teaching and learning of computer programming within a blended learning environment.

3.2.2 Behaviourism Theory

Behaviourism describes one of the earliest learning theories used in the scientific explanation of animal and human learning. Proponents of this theory place much emphasis on the measurable changes that are visible in the behaviour of animals and humans as they relate to the environment (Anderson and Elloumi, 2004). To the behaviourist the human mind which is used in learning is perceived as a "black box," owing to the fact that its responses to environmental stimulus can be observed quantitatively although the individual mental processes taking place in the mind are overlooked (Dietinger, 2003). Per the behaviourist school of thought, learning is alteration that is visible in the observed behaviour of learners arising from environmental stimuli (Dağ and Geçer, 2009). As a result of this Skinner (1974) asserts that the focus of instructors should be more on the cause-and-effect relationships that could be established by observation rather than inner processes which are not possible to identify. Mödritscher (2006) therefore posits that the behavioural theory of learning claims that only the behaviours that can be observed determine whether or not a learner has learnt a subject and not merely what is absorbed in the mind

Early computer learning systems were designed based on a behaviourist approach to learning. Teaching and learning of computer programming language according to this theory aids student's skills development to ascertain effective learning. According to Anderson and Elloumi, (2014) there is therefore a need for the instructors to provide a detailed learning outcome that shall form the basis for assessing learning behaviours. The teacher therefore has a duty to regulate the environment through the application of positive reinforcement to inspire desired behaviour as well as negative reinforcement and punishment to decrease undesired behaviours or responses (Shuell, 1986).

Pintrich (2000) in Boekaerts, Pintrich and Zeidner (2000) posit that behaviourists believe that the role of the student is to carry out what they are expected to do, in order to realize intentional or unintentional rewards. In order to realize this the learning approach, requires regular testing and examination to be sure the learning outcomes that reflect the particular behaviour are achieved (Dağ and Geçer, 2009).

3.2.3 Cognitive Learning Theory

Unlike the behavioural theories which emphasized behaviour changes as proof of learning, the cognitive theory according to Schunk (1989) focuses on the way and manner knowledge is acquired, developed, represented and remembered in the human mind. From the perspective of the cognitive theory, learning is an active process and learners must be involved in the learning process. Woolfolk, Davis and Anderman (2013) and Shuell (1986) indicated that the learning process of learners is usually influenced by their prior knowledge and experiences. According to the cognitive theory that stresses on the science of how the mind is used to store and process information, memory is divided into three main components which are sensory register, short-term memory, and long-term memory (Kay and Kibble 2016).

The first part of memory, which is the sensory register is responsible for receiving stimulus from the environment. This information is coded and stored briefly and this quickly gets lost if not attended to. Should it happen that the learner concentrates on the stimulus, the information is then sent to the short-term memory component of the memory (Anderson, 1983). The length of time this information is stored in this section is a function of how many rehearsals and repetitions the mind deploys this stored information in the execution of tasks (Atkinson and Shiffrin, 1968). With constant application of such information, it is then stored in the long-term memory. This section stores information all through the life of the individual. However, the ability to retrieve information from memory depends on how often such information is utilized (Kay and Kibble 2016).

Applying the cognitive theory to computer programming teaching and learning, learners must be very active and concentrate to be able to pick the initial stimulus by the sensory register that is crucial for understanding and storage in the short-term memory. Students must subsequently keep repeating, coding and rehearsing the programming language and applying it in the execution of tasks for the knowledge and skills to be stored in the long-term memory and be retrieved with minimal effort.

3.2.4 Social Constructivism Theory

According to Laurence and Margolis (1999) the work of Vygotsky is often aligned with the social constructivism movement. Subban (2006) argues that Vygotsky explored learning from the context of the social processes in which learners are able to develop with the aid of the intellectual capacity of those around them. The implication is that learning is a process in which learners are able to acquire knowledge and skills through interaction with others as well as cultural tools (Wood and Bandura 1989).

Subban (2006) indicated that from the standpoint of Vygotsky, knowledge comes from two main dimensions which are the interpersonal (external level) and the intrapersonal (internal level). Subban further asserted that it is not possible to attain the intrapersonal dimension of knowledge until the interpersonal aspect is achieved.

According to the social constructivism theory semiotic mechanisms such as cultural tools, language, processes, art, maps, and technology mediate on the social and individual functioning and connect the external and the internal, the social and the individual. While individuals can learn without the support or assistance of an expert, the social constructivists believe that learning can be best achieved when the learner receives support and guidance from others with high levels of knowledge and expertise.

In the context of programming, rather than focusing on performance, teaching and learning based on the social constructivist approach emphasises building the capacity of students to learn under the support and guidance of others (peers, teachers, colleagues, etc.) who are more knowledgeable to enable the students to gain in-depth understanding and skill. According to Adams (2006) this learning takes place within the context where students are afforded the opportunity to use appropriate cultural tools, technology and language.

3.3 Theoretical Framework of the Study

Grant and Osanloo (2014, p.13) state that *"theoretical framework is the "blueprint" for the entire dissertation enquiry. It serves as the guide on which to build and support your study, and also provides the structure to define how you will philosophically, epistemologically, methodologically, and analytically approach the dissertation as a whole".*

Yin (2004) posits that developing a theory is critical prior to the elicitation of data towards a research. Gregor (2006, p.614) asserted that *"in the scientific discipline, a body of knowledge together with conjectures, models and frameworks is encompassed by the word theory".* The theoretical framework adopted for this study is the Community of Inquiry (COI) model as applied in the context of blended learning by Garrison and Vaughan (2008, p.8).

Garrison and Vaughan (2008) are globally recognized by most researchers for their contribution towards the revolutionary activities in HEIs in the domain of "educational technology" and "educational process". Garrison and Vaughan (2008) further indicated that "reflection and discourse" are very essential compositions that cannot be separated from effective teaching and learning in HEIs.

Several researchers such as Chew et al., (2008); Anderson *et al.*, (2001); Szeto, (2015); Cavalcante, Riberas and Rosa, (2016) also agreed that the Community of Inquiry (COI) model (Garrison, Anderson and Archer, 2000) is a very effective model for implementing blended learning in higher education. A critical analysis of the Garrison, Anderson and Archer (2000) framework on the community of inquiry generates assurance that the model can be used in both face-to-face and online teaching and learning. Also, it can be seen clearly that the model has a component that supports students' learning engagement and active learning. Figure 3.1 depicts the conceptual framework adapted for this study.

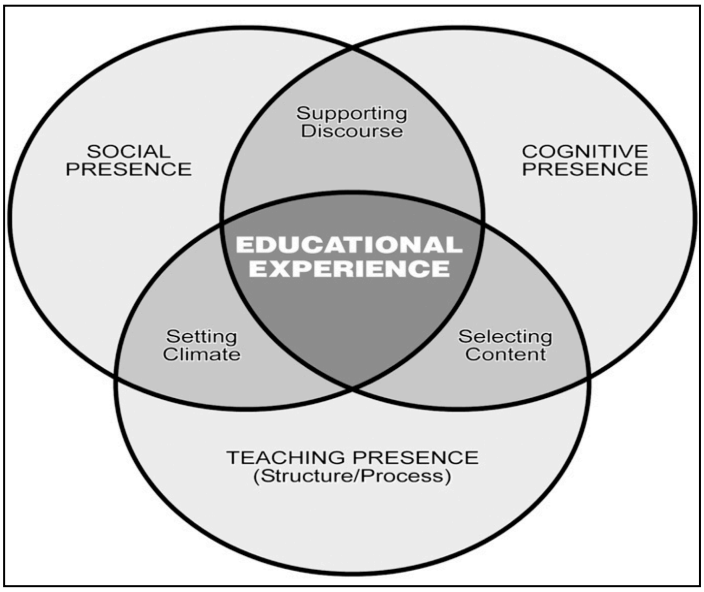


Figure 3.1: The Community of Inquiry Model, (Garrison, Anderson and Archer, 2010 p.2)

Further reflections on the Garrison, Anderson and Archer (2010) community of inquiry model also indicated that the COI framework gained its root in the study of Dewey (1916) on the constructivist theory of education (Grarrison and Vaughan, 2008, Garrison, Anderson and Archer, 2010). The constructivist theory has been used extensively even in the domain of computer science education (Kanuka and Garrison, 2004; Wulf, 2005; Harasim and Harasim, 2018; Suhendi, 2018). This is because Dewey's (1916) theory of education centres on not just gaining knowledge or memorizing things taught in school but rather gaining knowledge that seeks a practical exposure, and a learning journey that can aid in solving real-life problems. Because of this, it is undoubtedly noted that the constructivist theory of learning as employed in the COI framework is not a misplaced agenda at all.

The COI framework is based on the inquiry into learning process (Grarrison and Vaughan, 2008), hence learning and knowledge is acquired through social collaboration and interactivities. This affirms the core reasons why Garrison and Vaughan (2008, pp.15) indicated that "*education defined as a process of inquiry goes beyond accessing or even assimilating information. Inquiry joins process and outcomes (means-end) in a unified, iterative cycle. It links reflection and content by encouraging students to collaboratively explore and reasonably question the organization and meaning of subject matter*."

This supports why instruction of computer science programming should not be theoretically taught but rather engage the students actively. Applying the constructivist theory with the COI framework supports the recent publication of Harasim (2018 pp. 2) asserting that "C*onstructivist theory requires that we turn our attention by 180 degrees, we must turn our back on any idea of an all-encompassing machine which describes nature and instead look towards all those wonderful, individual living beings - the learners - each of whom creates his or her model to explain nature. If we accept the constructivist position we are inevitably required to follow a pedagogy which argues that we must provide learners with the opportunity to: a) interact with sensory data, and b) construct their world*".

The COI model indicates a complete teaching and learning process of inquiry. The three key dimensions used in this educational process as indicated in Figure 3.1 include: "Cognitive Presence", "Social Presence" and "Teaching Presence" (Garrison, Anderson and Archer, 2010).

The discussions on the cognitive presence, social presence, and teaching presence on the COI model are explained in the next sections.

3.3.1 The Cognitive Presence

According to the study of Garrison, Anderson and Archer (2010) the cognitive presence as applied in the COI framework is the basic component of the framework that enables the exchange of communication and information, testing of concepts supporting the discourse and content needs for teaching and learning. As indicated earlier on Dewey's philosophy of learning, the cognitive presence is achieved through "Practical Inquiry (PI)" based on reflective considerations of learning. This is because Dewey's theory believed that teaching and learning are achieved through effective reflective analysis of the learners' learning journey (Swan, Garrison and Richardson, 2009).

In line with this, the cognitive indication on the COI model provides a developmental and dynamic approach where the learner develops throughout the learning process. This worthwhile process enables the student to develop critical thinking. One quality measure of HEIs is to produce students who are well-disciplined and well-furnished to exhibit the skills learnt in school to solve problems in the society. This critical thinking ability of the student has a close relationship with the cognitive presence indication on the COI model.

3.3.2 The Social Presence

The introduction of computer and Internet technologies has made information dissemination and communication among learning peers and teachers easier and more convenient. Garrison, Anderson and Archer (2010) assert that the social presence indications on the COI model are very important to enhance the teaching and learning experience. The social presence of the model aids the learners to communicate among themselves and with their facilitators, thus supporting collaborative and group learning. Also, the social presence aids in forming an emotional consciousness of belonging to a social community among the learners and the teachers (Garrison, Anderson and Archer, 2010; Szeto, 2015b).

Grarrison and Vaughan (2008) on the other hand, contend that there were missing connections that existed between the teaching and learning experiences on the COI framework. Because of this, there was the need to develop an “open communication” among learners and teachers to enhance the social presence in their learning community. Consequently, Garrison (2009) provided a strong relationship between the social presence and the purpose of the academic learning experience with the COI model. It was noted by Garrison (2009) that it is a core interest for the student to gain “social identity” which is a prime concern for learners to feel emotionally connected to the social community of the learning environment and not “personal identity” within the community. Thus, providing the learners with a wider range for communicating opportunity in an environment they can trust and consequently aiding them to develop their interpersonal relationships (Garrison, 2009).

Inferring from the studies of Garrison, Cleveland-Innes and Fung (2010), and Szeto, (2015a) on the social presence affirmed that there is a strong positive relationship between teaching presence (discussed in the next section) and cognitive presence. In other words, the social presence serves as a mediating paradigm existing between the teaching presence and cognitive presence.

3.3.3 Teaching Presence

The teaching presence aspect of the COI model by Garrison, Anderson and Archer (2010) establishes a well-defined construction of all attributes that contribute to effective teaching and learning. At this stage, the design and strategic direction to enable the teacher or instructor to deliver teaching and learning of the course content to the learners are defined. Garrison and Arbaugh (2007, p. 164) specified that the teaching presence component of the COI model is “*a significant determinant of student satisfaction, perceived learning, and sense of community*”.

In spite of the importance of the teaching presence on the COI model, there is a theoretical gap of consent based on the structure of its delivery in an effective teaching and learning environment. Specifically, it is the morphology of designing the course, facilitating the course and directing the course among learners (Garrison, Anderson and Archer, 2010). In other words, the teaching presence component depends on the calibre of students.

Also, there is a no distinctive pedagogic mediation in the teaching presence that uniquely circumscribes which approach to employ at a particular time within a learning community.

In fact, the COI model employed within the blended learning environment is seen as a suitable framework that underpins this study. However, there are a few research gaps such as; pedagogic approaches, curriculum designs, facilitation and directions on the COI model.

The blended learning approach as propounded by Garrison and Vaughan (2008) propounded fits in the learning environment used to teach computer programming in the three selected universities participating in the study. Thus, the approach can be applied in both the face-to-face and online learning engagements.

3.4 Conceptual Framework of the Study

Conceptual modelling is the approach of "*identifying, analysing and describing the essential concepts and constraints of a domain with the help of a diagram*" (Guizzardi, et al., 2002, p.69). Also Pilkington and Pretorius (2015, p. 99) posit that a conceptual model or framework defines a "*formal structure or a representation about some aspect of the real world with the aim of allowing that domain to be better understood and communicated".* Hence, the researcher developed a conceptual framework to guide the study towards the development of the ABLA framework.

3.4.1 Development of the Conceptual Framework

The works of Dewey (1948) and Shields and Tajalli (2006) affirm that a conceptual framework is a map designed to direct a research to a destination, i.e. connecting a research from existing literature to a current study to solve a real problem. Dewey (1948) posited that a conceptual framework is needed to guide the research from literature, data gathering, data analysis and contribution of an inquiry. The conceptual framework serves as route-finding tool to link existing studies to the "experience or the experiential world" (Shields and Tajalli, 2006, p.416)

Also, the research conceptual framework provides a means to make reflections on previous studies, classify it and link to the problem of the study towards the data gathering, analysis and contribution of the study. That is, linking existing concepts to solve the research problem. To develop the conceptual framework of the study, the researcher developed a 2-phase process to uncover the research questions of the study. Figure 3.2 depicts the phases used in developing the conceptual framework of the study.

The conceptual framework of the study focused on the following components towards building a robust framework for activity-based learning in the instruction of computer programming.

1. Community of Inquiry - This consists of Teaching Activities, Cognitive Activities and Social Activities
2. Student Activities and Engagement - This includes engagement before the classroom, in the classroom and after the classroom
3. Feedback and Assessment - This constitutes both the peer and lecturer feedback. The assessment is both formative and summative
4. Curriculum Mediations - This consists of a developed curriculum that can be used to teach computer programming using an activity-based learning approach within a BL environment.
5. Learning Gains and Satisfaction – This constitutes all the constructs that contributes to significant positive impact of learning programming on students.
6. Students' Academic Performance and Programming Skills

Attempting to contribute to the body of knowledge, the researcher paid keen attention to develop a theoretical, practical, and policy framework for this study, hence, the findings from this research were derived from the evidence that emanated from this study to establish the philosophical contributions to knowledge. In relation to this study the researcher considered practical constructs that could contribute to an effective teaching and learning of computer programming among HEIs, hence the development of a conceptual framework to guide the construction of the ABLA framework. Before the construction of the conceptual framework, the researcher developed a schematic process for designing the conceptual framework as indicated in Figure 3.2 below.

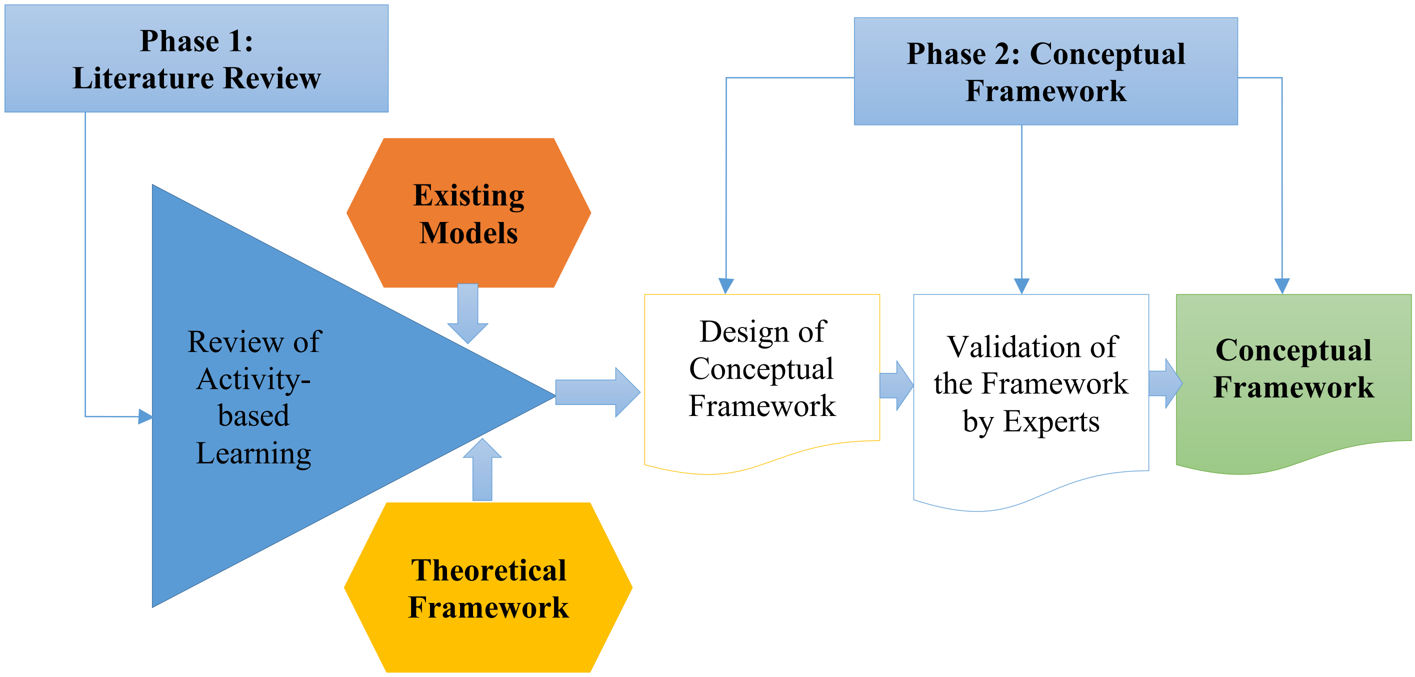


Figure 3.2: Schematic Processes for Designing the Conceptual Framework, Author (2019)

The first phase of the conceptual framework was developed from review of existing literature as discussed in Chapter 2 of this study. The various learning approaches and models used in teaching in higher education within blended learning environments were reviewed and the constructivism learning theory was adopted for the study.

The second phase is the construction of the conceptual framework using information gathered from the literature to unravel the research problems. Various components or factors that could contribute to effective teaching and learning of computer programming within a blended learning environment were considered and employed to design the conceptual framework. Correspondingly, the researcher took advice from experts who in this case are the supervisors and colleagues who are well versed in HEIs. The researcher's own professional experiences also helped to validate the conceptual framework.

Since the processes intended for the development of the conceptual framework have been outlined, there was a need to outline the compositions and factors that contributed towards the construction of the conceptual framework. In the next section (i.e. Section 3.5) the researcher explains the elements of the conceptual framework based on existing literature, the researcher's own professional experiences and direction from the supervisors of this research.

3.5 The Conceptual Framework

To begin with the construction of the conceptual framework, a thorough study of teaching and learning within a blended learning environment was conducted and the theoretical framework adopted for the study was discovered. The Community of Inquiry Framework as propounded by Garrison, Anderson and Archer (2010) served as the basis for the development of the conceptual framework for the study as depicted in Figure 3.3.

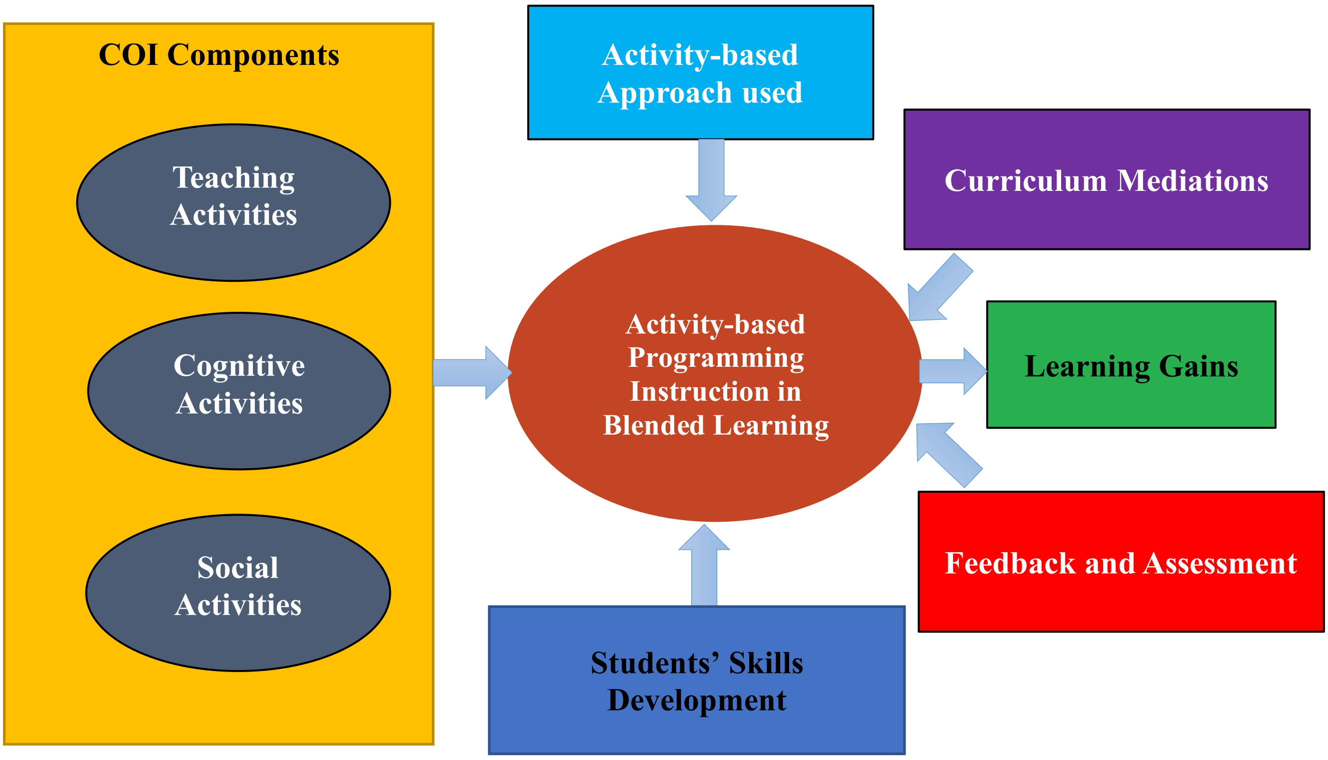


Figure 3.3: Conceptual Framework of the Study, Author (2019)

3.5.1 Community of Inquiry Component

The community of Inquiry (COI) model as indicated by Garrison, Anderson and Archer (2010) and applied within a blended learning environment consists of the teaching presence, cognitive presence and social presence. Hence, the researcher saw its relevance in the teaching and learning of computer programming within the context of the teaching activities, cognitive activities and social activities in the COI model.

1. **Teaching Activities**

   The teaching activities aspect of the framework establishes an effective approach for teaching and learning of computer programming. The design and strategic direction to enable the teachers to deliver teaching and learning of the course content to the learners is defined. The teacher or the instructor develops the teaching and learning approach, uploads course contents and resources, conducts assessments and grants feedback to students (Garrison and Vaughan, 2008).

2. **Cognitive Activities**

   The cognitive activities on the model operate as the central focus that establishes comprehension levels among the students. This phase of the COI model enables effective delivery of the course content to the students as they learn various concepts supporting computer programming. As indicated earlier on Dewey's philosophy of learning, the cognitive presence model is achieved through "Practical Inquiry (PI)" based on reflective considerations of learning. This is because Dewey believed that teaching and learning are achieved through effective reflective analysis of the learner's learning journey. In view of this the cognitive activities on the framework present a developmental and dynamic approach where the learner develops thought and critical thinking capabilities that can aid the student to solve problems.

3. **The Social Activities**

   The social activities aspect on the framework is relevant to enhance teaching and learning experience of computer programming using different social tools. By default, the LMS platform used provides a social component that enables students to learn within the blended learning community via collaborative learning and forum discussion. The social activities of the model aid the learners to communicate among themselves and the facilitators, thus supporting collaborative and group learning through forum discussions, and online chats. This as a result aids in forming an emotional consciousness of belonging to a social community among the learners and the teachers (Garrison, Anderson and Archer, 2010; Cripps, Jacobs, and MacCallum, 2020; Szeto, 2015b).

   Furthermore, the studies of Garrison, Cleveland-Innes and Fung (2010); and Szeto, (2015a) on the social activities' indications of the framework affirmed that there is a strong relationship between teaching activities and cognitive activities. In other words, the social activities serve as a mediating paradigm existing between the teaching activities and cognitive activities. This is because teaching and learning are seen as social endeavours.

3.5.2 Activity-based Approach Component

Student engagement is an essential factor for HEIs to promote effective academic performance, communication skills, self-motivation and sense of belonging and motivation among students (QAA, 2018; Nel, 2017). Hence, the activity-based approach component of the framework intends to build students' learning experiences using activities to solve real-world problems using computer programming concepts and skills. This approach is tailored towards the use of a Technology Enhanced Learning approach to engage students beyond the classroom environment and in the classroom to enhance teaching and learning seamlessly.

Some applications that support the teaching and learning delivery by actively engaging students include the use of social media tools like WhatsApp, Twitter, Zoom, Skype, Adobe Spark, Screencast O-matic and the LMS platform (Nouri, 2016). The activity-based approach could be in the form of Problem-based learning, quizzes, group discussions, projects, videos, field trips, etc. The rationale is to aid the teacher to gain an opportunity to engage with the students by assessing the academic growth of the students' learning journey. (Gilboy, Heinerichs, and Pazzaglia, 2015; Betihavas et al., 2015).

3.5.3 Learning Gains

The goal of effective teaching and learning is for the student to gain reasonable knowledge that can be applied in a useful manner to solve real world problems (Millheim, 2012). In the context of teaching and learning of computer programming within a blended learning environment using an activity-based instructional approach, it is expected that students perform well academically and also gain skills for the job market. The conceptual framework of the study highlights and measures the learning gains and satisfaction level of the students in activity-based computer programming courses within a blended learning environment. Hence, the researcher took interest to integrate Maslow's (1944) hierarchy of needs which is known to be one of the most recognized theories when it comes to expressing the needs of humans. As a result of that the learning satisfaction component on the model is used to enhance the motivations, learning gains and satisfaction levels of students' learning experience and consequently increase their performance levels (McLeod, 2007; Chew, Jones and Turner, 2008; Freitas and Leonard, 2011; Millheim, 2012).

3.5.4 Curriculum Mediations

Literature has confirmed that learning programming remains very difficult among most students in higher education institutions globally (Ahadi, Vihavainen and Lister, 2016; Yacob and Saman, 2012; Kanaparan et al., 2014; Sarpong, Arthur and Amoako, 2013; Lister et al., 2004; Wiedenbeck, LaBelle and Kain, 2004). Based on the prevailing challenges faced by most students in programming courses, the researcher indicated in the second research question of the study to explore curriculum mediation that can enhance computer programming instruction, i.e. "What curriculum mediation based on pedagogic approaches could support activity-based learning for instructing computer programming in HEIs?" The relevance of exploring curriculum mediation for teaching and learning programming is vital to quality and world-class education for computing degrees among HEIs.

Previous studies by The Joint Task Force on Computing Curricula (2013); Ngwenya and Futcher (2020), Ahadi, Vihavainen and Lister (2016); Robins, Rountree and Rountree, (2003); Kinnunen and Malmi, (2008) affirmed that the curriculum of computer programming instruction needs critical attention and must be improved to enhance students' programming experiences. As a result, the attention and improvement of the existing programming curriculum include:

1. The redesigning of the programming course structure (Oliva and Gordon, 2012),
2. Higher education management supports (Pinar, 2012),
3. Application of technology enhanced learning tools (Powers, et al., 2006; Derus and Ali, 2012) and
4. Effective pedagogic and assessment approaches for instruction (Robins, Rountree and Rountree, 2004).

The researcher agrees with the study of Jenkins (2002, p.54) who emphasizes that "*If students struggle to learn something, it follows that this thing is for some reason difficult to learn*". This strong claim calls for the attention for this study to explore an effective curriculum mediation to enhance computer programming instruction among students within a blended learning environment since the current curriculum used among most HEIs has less effect on students' performance and programming skills.

3.5.5 Feedback and Assessment

Assessments and feedback are essential elements in the delivery of teaching and learning (Boud and Associates, 2010; Carless et al., 2011). This is because the assessment and feedback component of teaching and learning engages learners to be productive in their learning journey. Studies have affirmed that there are no explicit methods of conducting a quality assessment. This further explains that assessment activities must conform to best practices, standards, and detailed assessment criteria (Boud and Associates, 2010; Carless et al., 2011; Fluckiger et al., 2010; Gibbs and Simpson, 2004; Gilbert et al., 2011; Miller et al., 2010; Sadler, 2010).

The feedback activities are considered as a progressive activity that a student acts upon within a particular period of time (Boud and Associates, 2010; Carless et al., 2011; Fluckiger et al., 2010; Gibbs and Simpson, 2004; Gilbert et al., 2011; Miller et al., 2010; Sadler, 2010). The feedback activities on the other hand inspire students to reflect on their individual academic journey and otherwise make amends, improve or maintain the learning experiences (Lew, Alwis, and Schmidt, 2010; Nicol, 2008).

Price and Kirkwood (2011) asserted that assessment and feedback are effectively carried out following the integration of technology to enhance assessment. Among the TEL tools that support effective and quality assessment and feedback include Turnitin, Moodle, Socrative, Screen Cast O-matic, Sakai, etc. For example, Mohanty and Vohra (2006) and Price and Kirkwood (2011) posit that higher education institutions must employ the use of Turnitin to check plagiarism or similarity index of students' assessment submissions. These tools can be used to perform both summative and formative assessments (Prince and Kirkwood, 2011). Lecturers can further give instant feedback to the students using text, audio or recorded video to grant feedback to the student(s) via the LMS. Tools that support audio-visual feedback are Adobe spark, Screen Cast O-matic, Socrative, etc.

3.5.6 Students' Skills Development

Several researchers such as Freeman et al., 2014; have asserted that an activity-based approach of teaching affects students' performance positively in terms of their skills. On the other hand, the research after examining previous studies on the abysmal performance of students in computer programming courses (Greyling et al., 2006; Tie and Umar, 2010; Sarpong, Arthur and Amoako, 2014) saw the need to measure students' performance based on the activity-based approach employed with the blended learning environment. Hence, the researcher saw the need to examine students' performance and skills on the framework.

3.6 Conclusion

This chapter discussed the learning theories, theoretical and conceptual frameworks that underpin this study. The conceptual framework developed for this research outlined the key components inferred from the theoretical framework, literature review, expert opinions and personal experiences of the researcher. Hence, the conceptual framework employed the integration of the COI model and students' engagements, feedback and assessment, curriculum mediations, establishing learning gains and satisfaction and finally measuring the framework on students' academic performance and programming skills acquired. The researcher also took care to validate the framework hence, experts' opinions were also taken and added to the construction of the conceptual framework.

CHAPTER 4

RESEARCH METHODOLOGY

4.1 Overview of the Methodological Approach

Research methodology, as defined by Creswell and Poth (2018, p.320), "*involve(s) the forms of data collection, analysis, and interpretation that researchers propose for their studies.*"

The main purpose of this study is to develop a framework for an activity-based computer programming instruction within a blended learning environment in HEIs in Ghana. The study focused on an activity-based approach for teaching computer programming. Computer programming was chosen because, the researcher believes that integrating activity-based learning in computing sciences education is a factor for enhancing technological innovation, skills development and employability of undergraduate students. Also, the researcher believes that employing the activity-based approach for instructing computing sciences' programmes will help to harness the foundation of technology integration in teaching other disciplines such as business, engineering, agriculture, etc.

From the literature review, the researcher took note that there should be a paradigm shift in the teaching practices of computer programming. By way of establishing a framework for teaching computer programming in HEIs, the adoption of an action research approach was implemented for this study within a blended learning environment.

This chapter discusses the researcher's worldview and its relevance to the study. The action research design and the mixed method approach was used to conduct this study and the details of how it was used are discussed in this chapter. Moreover, the chapter discusses the research population, sample, and data collection techniques for the used for this research. The techniques for conducting the reliability and validity of the study have been discussed in this study.

4.2 Philosophical Perspective of the Research

In order to properly plan the research, the researcher needs to adopt a philosophical worldview and assumptions on the study (Creswell, 2014). In view of this, the approach that relates to the worldview of the researcher and its significant practical implications in the world is the constructivist worldview. The term worldview is "*a basic set of beliefs that guide action*" (Guba, 1990, p. 17). The researcher sees the various worldviews as a universal orientation about the world and how a study needs to be perceived and conducted.

Taylor and Bogdan (1998) observed that, there are two main theoretical worldviews in social science researches, namely positivism and phenomenology. They also underscored that the constructivist theory can be applied in teaching and learning, thus its adoption for this research. The constructivist theory alludes that the researcher actively serves as an agent through acquisition of knowledge within a reflective process.

Creswell (2014) indicated four different worldviews of a research. These include, post-positivism or positivism, constructivism, pragmatism, and advocacy or participatory worldviews. The assertions of Taylor and Bogdan (1998) and Creswell (2014) have shed similar light on the epistemological and ontological orientations of research that relates to qualitative, quantitative or mixed approaches. Morse (1991) also asserted that both qualitative and quantitative approaches co-exist but with different scope and processes and thus together form a triangulation for data validation.

The researcher's epistemological orientation was the constructivist paradigm as Creswell (2014) and Taylor and Bogdan (1998) asserted. The justification for this chosen paradigm was that, as a constructivist, the researcher started by constructing his own knowledge of how he sees the world. Thus, the researcher theorized the study, accepted data that was either true and supported the study or refuted the theory. The researcher then constructed knowledge and consequently reflected on the knowledge acquired (Bereiter, 1994). The constructivist researcher makes amendments before the final test is made. Similarly, the researcher studied and explained the social viewpoint and the experience of the individuals who were unswervingly involved in the social and academic processes in the teaching and learning of computer programming within a blended learning environment in HEIs.

Creswell (2014) asserted that as a constructivist researcher, it is required to have an unwavering viewpoint that will aid to conclude the research outcomes and effects. In other words, the researcher reflected on the necessities to discover and assess the causes that influence results through knowledge acquisition as applied in an experimental research with the students' learning experiences. As a result of the constructivist worldview, the researcher was guided by different sets of ideas during the experimental, observation and knowledge construction stages in different computer programming classes at three different universities in Ghana and finally tested for possible results to construct an Activity-Based Learning Approach (ABLA) framework.

Thereafter, the researcher observed carefully the effects of the proposed ABLA design on students and lecturers and consequently developed a numerical measurement to analyse the study. The adopted constructivism philosophical worldview influenced the design of the adopted framework as discussed in the next section.

4.3 The Justification for Adopting the Constructivist Theory

Upon personal reflection of teaching and learning in HEIs, it is noted that there is a shortfall of teaching and learning among teachers and students in terms of delivery and pedagogic approaches used. The reason being that in most cases instructors just convey knowledge to students without students constructing their own knowledge. Students mostly produce what the teacher has instructed (Bada, 2015). Critically, learners must construct their own knowledge, ascertain new information from old information and make amends to their learning experience as they acquire knowledge (Ali, 2005; Harasim, 2018).

Unequivocally, the constructivist worldview of research which enshrines that the researcher serves as an active agent in the process of getting understanding of the phenomenon best fitted for this research. This is because both teachers and learners who were actively engaged in this research constructed their own knowledge; the teachers guided the learners while the learners reflected on what they had been taught. The constructivist theory of learning was conceptualised way back in the histories of Dewey (1929); Bruner (1961); Vygotsky (1962) and Piaget (1980). These researchers have proposed that the constructivist approach of instruction and learning is based on the learner's ability to develop individual cognitive construction of knowledge. In other words, students are able to learn new things that have got effects on the things they previously know (Bada, 2015).

As Wulf (2005), Smart, Witt and Scott (2012) and Bada (2015) asserted, the constructivists have a strong judgement that teaching and learning is precious as a result of the circumstance or context within which knowledge is taught, the beliefs of the individual students as well as the attitudes of the students. In other words, the theory believes that by nature, human beings are able to make meaning of something, construct knowledge and learn based on one's individual experiences. Also, it is believed that the constructivist theory enables students to actively get engaged by applying their prior knowledge, personal experiences, and personal reflection of thought as acquired.

Accounting for the relevance of the constructivism theory in research, it is noted that the worldview fits best for this research as compared to other worldviews such as positivism/post-positivism, participatory and pragmatism (Creswell and John, 2018). The relationships between the worldviews, designs and methods are indicated in Figure 4.1 below.

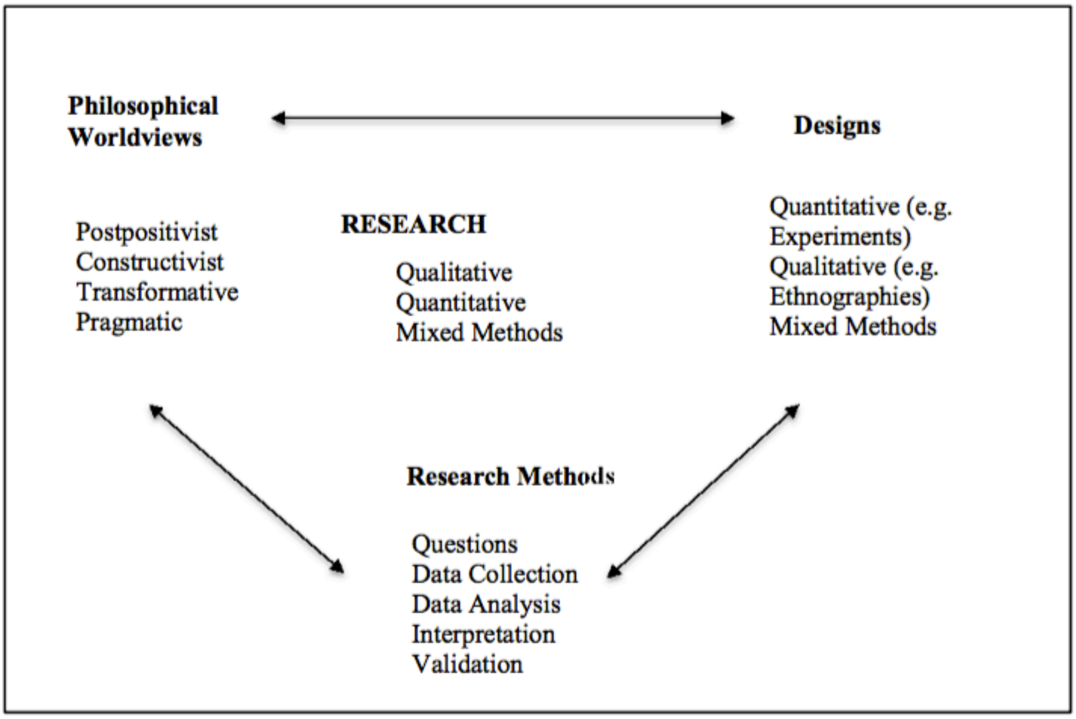


Figure 4.1 Different Worldviews of Research and Methods Used, Source: Creswell, (2014).

Also, it is relevant as a constructivist researcher to develop a suitable theory or a framework to understand the context in which computer programming is taught in HEIs and also to promote learning experiences of learners through a social collaborative blended learning environment. As a result of this, the researcher's contribution to the research for developing an ABLA framework for computer programming instruction in a blended learning environment is novel as a constructivist researcher.

4.4 Action Research (AR) Design

This study was centred towards the use of AR design to develop an activity-based instructional approach for teaching computer programming within a blended learning environment. The concept of AR has been established as a form of a dynamic study of social change where the research provides a natural means of participating, learning, and studying at the same time (Nel, 2017; Dick, 2002). According to De Villiers (2005b), action research methodologies provides a wide range of data gathering sources that offer unique prospects for integrating research into university teaching and learning. Furthermore, they offer efficient techniques for forming learning communities that bridge conventional barriers between educators and students while also improving professional development and pedagogy in interdisciplinary contexts. (Nel, 2017; Baskerville, 1999; Cohen, Manion and Morrison, 2000, and Chen and Zelinsky, 2003). In order to conceptualize real world scenarios, AR is used to relatively provide a small-scale intervention that mimics the functionality of the real world. This approach is relevant to the study in the academic community where several iterations of studies and investigations are conducted to establish an intervention for a problem (De Villiers, 2005b; Herington and Weaven, 2008). The action research approach depicts the philosophy of the researcher by recognizing the fact that it is a best practice when the lecturer who is seen as the expert in the teaching practice is also involved in the research.

According to the studies of Baskerville (1999); Herington and de Villiers (2005b); Herington and Weaven (2008), AR is participative, iterative, introspective and qualitative as indicated in Figure 4.2. As a result of adopting the AR approach, the researcher designed the research iteratively by participating, making reflections and performing a cycle of activity until the ABLA framework was developed.

Additionally, de Villiers (2005b) alluded to the repetitive and introspective nature of an action research that is well illustrated by the model in Figure 4.2 below as a series of cycles pointing to solving a particular problem.

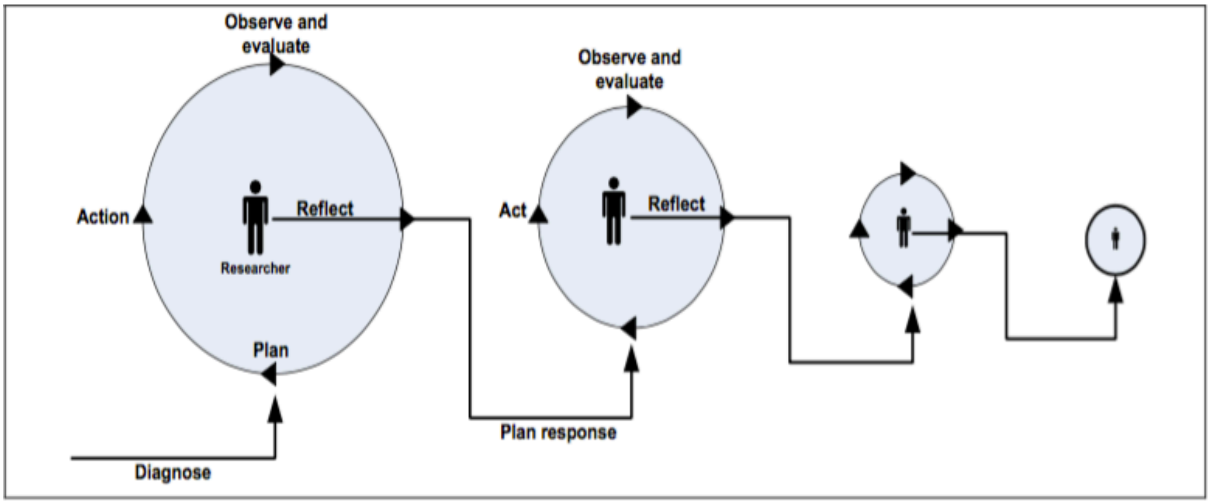


Figure 4.2: Action Research Model, Source: De Villiers (2005b)

The justification for choosing an AR approach was determined by the fact that the researcher could actually plan the research, act upon it, make a series of iterative observations, make evaluations of the study and perform reflections. This method also aided the researcher to perform the principal role for actively participating in the research.

Furthermore, the AR approach aided the lecturers and students to gain confidence, learn from each other and consequently helped both the learners and instructors to improve the teaching and practice of computer programming instruction within a blended learning environment. The study of Chee (2008) supports the argument that AR flexes an academic inquiry and thereby improves teachers' and students' teaching and learning experiences. Furthermore, Cohen, Manion and Morrison (2005) asserted that AR methodology is applied to improve problematic instances of academic achievements.

All the assertions associated with AR by Baskerville (1999), Herington and De Villiers (2005b), Herington and Weaven (2008), Cohen, Manion and Morrison (2005), and Chee (2008) were considered very relevant for the researcher in order to contextualize the study to suit the instruction of computing programming within the context of higher education institutions in Ghana.

4.4.1 Action Research Design Processes Used

In this section, the researcher outlines the steps that aided to define the logical focus of the action research implemented for this study, thus the evident steps that are required to address the research questions. Consequently, the researcher developed a seven-stage research process to undertake this research as indicated in Figure 4.3.

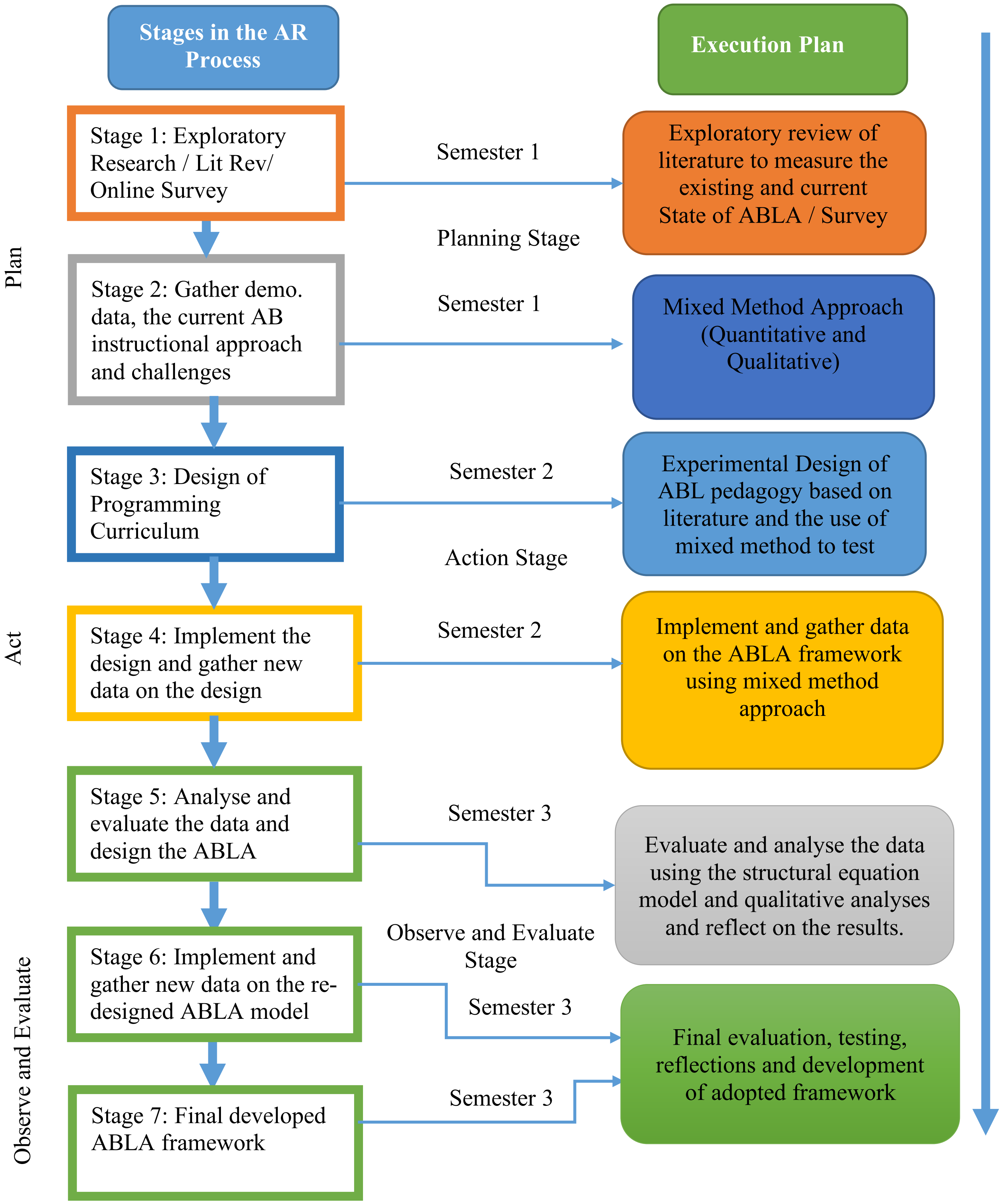


Figure 4.3: Action Research Design Process Model, Author (2019)

The rationale for developing the action research design processes was to enable the researcher to plan the research, act upon it, perform a series of iterative observations and evaluations of the study and consequently perform reflections of the study. The researcher took a keen attention to employ the processes in order to elicit data from participants and thus aided in the development of the ABLA framework for teaching and learning programming.

Stage 1: Exploratory Literature Review

The first stage of the action research comprises a detailed exploratory literature review on educational concepts in HEIs, the philosophical assumptions of activity-based learning approaches, activity-based learning, blended learning (BL), and smart learning. In this phase, the researcher addressed the theoretical basis that formed the major foundation of the research. This stage formed the basis for the next six stages as indicated from the chapters two and three of this research and consequently contributed towards answering the first research question, i.e. what are the current activity-based learning approaches in computer programming instructions within a blended learning environment? This activity occurred before and during the planning stage of the first semester of the action research stages.

Stage 2: Gather Demographic Data, the Current AB Instructional Approach and Challenges of Teaching Programming

In the second stage of the research, the researcher gathered the demographic information of the respondents among the three HEIs. The data on current activity-based instructional approach and challenges of teaching computer programming within a blended learning environment was also collected using an online Google form. This activity occurred during the first semester of action research stages. The second stage also constitutes part of the planning stage of the action research.

Stage 3.0: Design of ABLA Framework

The researcher at the third stage acted and designed a blue print of the ABLA framework that emanated from the conceptual framework of the study and existing literature of teaching computer programming within a blended learning environment. This approach was established based on extensive study on the conceptual framework and inputs from colleagues who teach computer programming among the three HEIs. Questions like Why ABLA? What are compositions of the ABLA framework? What could be the learning gains or benefit of ABLA to students and lectures? were addressed. Among these were the key areas that guided the researcher to design the very first framework for an ABLA framework. This activity occurred during the second semester of action research stages and data was collected on the various components.

Stage 3.1: Composition of the ABLA Framework

The composition of the framework includes the integration of:

1. Adoption of the Community of Inquiry Model (COI) within a blended learning environment: This includes the teaching activities, cognitive activities and social activities as employed by Garrison, Anderson and Archer (2010).
2. Student's engagement through learning activities (Roffe, 2002; Nel, 2017; Schindler et al., 2017).
3. Teaching and Learning gains (Akyol and Garrison, 2011; Milheim, 2012).
4. Feedback and Assessments i.e. (summative and formative) (Chew, Jones and Turner, 2008; Olsson and Mozelius, 2016).

Stage 3.2: Approach to ABLA Framework Construction

To enhance an effective teaching and learning experience among students in HEIs, an innovative approach needed to be adopted. Hence, the researcher developed the ABLA framework using the following processes:

Step 1: Adopt computing science or information technology curriculum approved by National Accreditation Board (NAB) in Ghana by the three higher education institutions selected for this study.

Step 2: Develop or adapt a course outline used by the instructors in selected schools.

Step 3: Develop the teaching strategies to implement activity-based learning with a well-defined timeline of activities.

Step 4: Integrate technology enhanced teaching and learning platforms or tools in teaching. This step contributed to the cognitive and social presence of the learner.

Step 5: Develop assessment strategies (formative and summative) and feedback strategy.

Step 6: Evaluate students' feedback and performance.

Step 7: Establish the first conceptual ABLA framework.

Step 8: Evaluate and assess the data for the three semesters and finally adopt the ideal ABLA framework.

Stage 4: Implement the ABLA Design and Gather New Data on the Design

At this stage, the researcher acted and implemented the design that was designed in Stage 3. In view of this, the researcher planned, acted, observed and reflected on the design iteratively to develop the new framework. To apply an appropriate design strategy, the mixed method approach was adopted to gather data for analysis and measure whether or not the proposed design pedagogy fitted into teaching computer programming in an activity-based learning environment. A web-based research survey, questionnaire administration, interviews and personal observations were employed to capture the data. This consequently aided the researcher to answer research questions 3 and 4. The evaluation and the questionnaire were developed based on existing frameworks discovered from the literature. The purpose of employing the mixed method at this stage was to triangulate the findings (Creswell, 2014, p.16).

Stage 5: Analyse and Evaluate the Data and Redesign the ABLA

At the fifth stage of the action research process, the researcher analysed, evaluated and reflected on the data derived from the designed ABLA framework following the mixed method approach. The structural equation model was used to establish the relationships and effects that exist among the variables. Thematic and content analysis was performed on the data derived from the qualitative data.

Stage 6: Implement and Gather New Data on the Re-designed ABLA Model

The researcher conducted a personal reflection on the findings and allowed other faculty members who are experts in computer science and teach programming to implement the model and grant feedback on the re-developed model in stage 5 through interviews and observations. At this stage, the researcher also gained good groundings to develop the full framework at stage seven following the iterative process after three successive semesters.

Stage 7: Adopted Framework

This was the last stage for the research. At this stage, the researcher re-designed the framework based on the evaluation and personal reflections conducted for three semesters.

4.4.2 Action Research Timelines

The researcher used the following timelines for the research as depicted in Table 4.1 below.

Table 4.1 Action Research Timelines

Stages	Activities	Dates	Data Sources
1 & 2 of AR Cycle. (Semester 1)	Gather data on current pedagogic approach in computer programming instruction for the first semester	30th September 2019 – 30th October 2019	Appendix B: Questionnaire for (Students) Appendix C: Students' Interview Guide.
3 & 4 of AR Cycle. (Semester 2)	Design of ABLA framework in the second semester	1st November 2019 – 15th December, 2019	Appendix D: Faculty Interview Guide on Curriculum Design
	Implement design	5th January 2020 – 1st April 2020	
5 & 6 of AR Cycle (Semester 3)	Gather data on students and faculty's experience on the framework after second semester	15th January – 20th February 2020	Appendix E: Questionnaire for Respondents (Students) – After Semester 2 Appendix F: Students Interview Guide on the ABLA Framework (Semester 3) Appendix G: Faculty Interview Guide on the ABLA Framework (Semester 3)
	Analyse and evaluate the data and re-design the framework.	1st April 2020 – 20th April 2020	Appendix E: Questionnaire for Respondents (Students) – After Semester 2
7 of AR Cycle	Perform reflection on the framework for the third semester	1st May 2020 – 1st June 2020	Appendix E: Questionnaire for Respondents (Students) – After Semester 2
	Final evaluation, testing and development of adopted framework	1st October 2020	

Source: Author's Timelines of Action Research (2019)

4.5 Research Hypotheses

The following research hypotheses were formulated to conduct the research;

1. H_1: The activity-based teaching and learning approach of computer programming in HEIs in Ghana within a blended learning environment include but not limited to case studies, quizzes, projects, group discussions and presentations, problem-based learning and concept mapping.
2. H_2: The new developed curriculum used in teaching computer programming within a blended learning environment supports activity-based learning.
3. H_3: The activity-based learning approach of teaching programming within HEIs in Ghana has a significant positive effect on students' learning gains.
4. $H_{3.1}$: The activity-based learning approach of teaching programming within HEIs in Ghana has a significant positive effect on students' cognitive development.
5. $H_{3.2}$: The activity-based learning approach of teaching programming within HEIs in Ghana has a significant positive effect on students' social activities and learning.
6. $H_{3.3}$: The activity-based learning approach of teaching programming within HEIs in Ghana has a significant positive effect on students' teaching activities.
7. $H_{3.4}$: The activity-based learning approach of teaching programming within HEIs in Ghana has a significant positive effect on students' engagement.
8. $H_{3.5}$: The activity-based learning approach of teaching programming within HEIs in Ghana has a significant positive effect on students' feedback and assessments.
9. $H_{3.6}$: Social activities have a significant positive effect on learning gains among students in HEIs in Ghana.
10. H_4: The developed framework has a significant positive effect on the activity-based learning approach in a blended learning environment for instructing computer programming.

4.6 Application of the Mixed Method Approach in the Action Research

The researcher applied the mixed method approach within the action research design method to solicit for both qualitative and quantitative data. Pilkington and Pretorius (2015) asserted that both qualitative and quantitative approaches can be used to conduct action research design. The following steps were used in analysing the quantitative data:

1. Design the questionnaire to collect the data.

2. Analyse data received from the online survey for semesters 1 and 2 and screen the data to gather the demographics, the current activity-based approaches used, the various challenges in teaching programming, etc., and measure the reliability and validity of the data to conduct the research.
3. Perform descriptive statistics and ANOVA test on the data.
4. Conduct an exploratory factor analysis to check for all missing values, outliers, and normality as to whether the data was appropriate for data analysis using Structural Equation Model (SEM) (Hair et al., 2014) was conducted.
5. Test the hypotheses of the research to answer the research questions.

On the other hand, the qualitative data included interviews, observations, and focus group discussions among students and lecturers while the quantitative data also includes questionnaire administration and online surveys of students and lecturers of HEI-1, HEI-2, and HEI-3 using blended learning approach for teaching and learning.

The following steps were used for the qualitative studies:

1. Design an interview guide
2. Administer and conduct the interview to gather data from students and lecturers
3. Perform focus group discussions
4. Conduct observations

The justification of these chosen methods is that both qualitative and quantitative research approaches and the ontologies and their epistemologies are closely associated in solving the research questions. In other words, the findings from the qualitative and quantitative data aided the researcher to establish judgements that contributed to the development of the ABLA framework.

Bryman (2012) and Cohen, Nanion and Morrison (2001) asserted that constructivism, epistemologies and realism ontologies are associated with the use of a quantitative approach of research. Also, a qualitative research approach is associated with interpretive or post-positivist epistemology and constructive ontologies. The researcher's approach of adopting the constructivism theory was not in a misplaced position for conducting both qualitative and quantitative analysis of data within an action research design. In order to triangulate and gather more informed and deeper understanding of the research questions, the researcher adopted the action research design, mixed method approach vis-à-vis the philosophical assumptions i.e. the constructivist theory of the study to unravel the research questions.

4.7 Research Approach

Creswell and Poth (2018, p.320) defined research approach as "*plans and the procedures for research that span the decisions from broad assumptions to detailed methods of data collection and analysis. It involves the intersection of philosophical assumptions, designs, and specific methods*".

The research approach adopted for this study was the inductive research approach. The study of Soiferman (2010) asserts that induction research approaches the research reasoning from a specific dimension to a general perspective. The justification for this chosen approach aided the researcher to generate different meanings from empirical reviews and data collected to establish relationships and patterns to develop a theory. Also, this approach aided the researcher to learn based on the experiences of the study. The inductive approach also permitted the researcher to use existing theories to formulate new theories (Saunders, Lewis and Thornhill, 2012).

Additionally, the researcher performed a sequential exploratory review of existing theories and frameworks on activity-based learning in teaching computer programming within a blended learning environment. The researcher employed the mixed method within the context of action research design. That is, employing both qualitative and quantitative research approaches (Creswell and Porth, 2018) to make arguments from experiences, observations, interviews, surveying, questioning, and participations among undergraduate students and lecturers in HEIs in Ghana. The HEIs selected for this study are represented as HEI-1, HEI-2, and HEI-3.

Primary data was collected from an online questionnaire, face to face interviews, observations, and focus group discussions. The secondary data was collected through empirical literature studies of similar researches for the teaching and learning of computer programming in HEIs. Both primary data and secondary data were gathered for analysis.

The study accepted primary data of the action research process and quantitatively or qualitatively analysed the data of students' and lecturers' views on the designed ABLA framework at the undergraduate level using SEM and thematic analysis. Online surveys and questionnaire administration were the main instruments to collect the quantitative data.

As a constructivist researcher, it was imperative to observe and interview respondents to construct knowledge for the research. The qualitative data were gathered through experiences, observations, interviews, focused group discussions and responses from undergraduate students and lecturers in programming classes. It is against these approaches that the researcher actively involved himself in the research and passed judgement through active engagement of students in teaching computer programming at the undergraduate level.

4.8 Research Design and Methods

Creswell and Poth (2018, p.320) defined research designs as "*types of inquiry within qualitative, quantitative, and mixed methods approaches that provide specific direction for procedures in a research study*". The study used an action research approach to perform reflection and iteration of students and lecturers' adoption and practice of activity-based learning in computer programming instruction within a blended learning environment.

The researcher employed the mixed method approach within the action research design approach. Creswell and Poth (2018, p.317) defined mixed method as "*an approach to inquiry that combines or integrates both qualitative and quantitative forms of research. It involves philosophical assumptions, the use of qualitative and quantitative approaches, and the mixing or integrating of both approaches in a study*".

The chosen action research design provided a comprehensive guideline for data collection in order to provide evidence to answer the research questions. The emphasis of the research design finally contributed to the development of the robust framework that can be used to teach computer programming using an activity-based learning approach in a blended learning environment.

4.9 Framework of Research Design and Methods for the Study

To answer the main objective of the study, various techniques were used in the methodology chapter to address the research objectives and the research questions respectively. The study employed different techniques in order to answer the research questions.

The driving force to employing different techniques for collecting the data and analysing the data are that there are different variables that are being observed with different purposes and assumptions, hence requiring different measurement techniques. The summary of the methodological technique employed for each research objective and research question is indicated in Table 4.2 below.

Table 4.2: Framework of Research Methodology Used for the Study

No.: OBJ	Research Objectives	Hypotheses of the Research	Data Source	Timelines	Types of Data
1	To ascertain the current activity-based learning approaches in computer programming instruction in HEIs within a blended learning environment.	H_1: The activity-based teaching and learning approach of computer programming in HEIs in Ghana within a blended learning environment include but not limited to case studies, quizzes, projects, group discussions and presentations, problem-based learning and concept mapping.	Primary data from an online questionnaire Interviews, focused group discussions	First Semester 30th September 2019 – 30th October 2019	Empirical data Quantitative and text from respondents (Qualitative)
2	To develop a curriculum mediation based on pedagogic approaches and application that could support activity-based learning for instructing computer programming for HEIs in Ghana.	H_2: The new developed curriculum used in teaching computer programming within a blended learning environment supports activity-based learning.	Data from system developed Interviews, focused group discussions	Second Semester 15th January – 20th February 2020	Rapid application Development Qualitative

3	To investigate the learning gains of an activity-based learning approach in a blended learning environment for students and lecturers in HEIs of Ghana.	H_3: The activity-based learning approach of teaching computer programming within HEIs has a significant positive effect on students' learning gains. $H_{3.1}$: Th activity-based learning approach of teaching computer programming within HEIs has a significant positive effect on students' cognitive development. $H_{3.2}$: The activity-based learning approach of teaching computer programming within HEIs has a significant positive effect on students' social activities and learning. $H_{3.3}$: The activity-based learning approach of teaching computer	Primary data from online questionnaire Interviews, focused group discussions	Second Semester 15th January – 20th February 2020 and Third Semester 1st May 2020 – 1st June 2020	Quantitative and Qualitative

		programming within HEIs has a significant positive effect on students' teaching activities (Instructional activities). $H_{3.4}$: The activity-based learning approach of teaching computer programming within HEIs has a significant positive effect on students' engagement (Skills development). $H_{3.5}$: The activity-based learning approach of teaching computer programming within HEIs has a significant positive effect on students' feedback and assessments. $H_{3.6}$: Social activities have a significant positive effect on learning gains.			

4	To develop a new framework that enhances an activity-based learning approach in a blended learning environment for teaching computer programming in HEIs.	H_4: The developed framework has a significant positive effect on activity-based learning approach in a blended learning environment for instructing computer programming.		Third Semester 1st May 2020 – 1st June 2020	Quantitative and Qualitative

4.10 Research Population and Sampling

4.10.1 Sampling Procedure

Sampling, as defined by Sekaran and Bougie (2010, p. 230), is the "*process of selecting a sufficient number of the right elements from the population, so that a study of the sample and an understanding of its properties or characteristics makes it possible for us to generalize such properties or characteristics to the population elements*". It is noted that it is practically difficult to study the entire population considering the selected participating institutions for the research. Hence, choosing a sample from a population makes the research practically easy to accomplish the study considering the time and cost involved to execute the research.

It is relevant in a survey research that involves participants and experts to give relevant responses to questions towards achieving the research questions. In pursuit of that, there was the need for the researcher to select and determine the sampling procedures for defining the sampling size that can accurately aid in obtaining the results and making judgement and generalization of the data received.

Correspondingly, the researcher employed the sampling procedure as suggested in the study of Sekaran and Bougie (2010). In other words, the researcher first defined the population for the research and secondly established the sample frame for the study for the action research process.

4.10.2 Population

Research population refers to the sets or groups of people whom research conclusions are drawn from (Babbie, 2013, p.115). In another context, Hair, Wolfinbarger, Ortinau and Bush (2010, p. 131) also stated that it is relevant to have a target population for a research hence they asserted that a target population is a "*complete group of elements (people or objects) that are identified for investigation based on the objectives of the research project*". The researcher agrees with Hair et al. (2010, p. 131) on the issue of getting a target population for the research to dichotomize between which elements within the population needed to be considered or not for participating in the research.

Hair et al. (2010, p. 131) indicated that choosing a targeted population aids in selecting the participants of the research based on their age, gender, beliefs, worldviews, locations, etc. In the context of this research, the researcher targeted students who were above eighteen (18) years and lecturers in the teaching and learning processes of computer programming within the three HEIs in Ghana.

Additionally, to effectively test the design of the ABLA framework from Stage 4 of the action research process and recursively perform reflections and evaluations, there was the need to involve people (i.e. students and lecturers) and consequently to solicit their views on the design.

An attempt to employ the mixed method approach with the selected universities as part of the population was not in a misplaced order for the researcher. Creswell and Poth (2018, p.317) described that "*mixed methods design involves the use of one or more core designs (i.e., convergent, explanatory sequential, exploratory sequential) within the framework of a single or multiple case study design. The intent of this design is to develop or generate cases based on both quantitative and qualitative results and their integration*".

The population targeted for this research constituted undergraduate computing students from faculties and departments of computing offering degree programmes, i.e. degrees in Computer Science, Information Technology, Information Systems, and Computer Engineering at HEI-1, HEI-2 and HEI-3 all based in Accra, the capital city of Ghana. Table 4.3 depicts the population and sample of the participating institutions selected for the study.

Table 4.3 Population of the Study

Institutions	**Entire Student Population (2019/2020)**	**Targeted Population Enrolled for Computing Degrees**	**Targeted Population of Teaching Faculty in CS and IT Departments**
HEI-1	9,015	660	5
HEI-2	10,169	620	5
HEI-3	4,500	600	5
Total	23,684	1,880	15

Source: Population and Sample of the Selected Universities, (2019) Hence, the entire targeted population for the study constituted one thousand, eight hundred and eighty (1,880) students among the three higher education institutions in Ghana. Also, the targeted population for the lecturers consisted of 15 teaching faculty members among the three HEIs.

4.10.3 Sample Frame

According to the study of Sekaran and Bougie (2010), after defining the research population for a study, the next phase is to determine the research sampling frame. Sekaran and Bougie (2010) asserted that sampling frame is "*a representation of all the elements in the population from which the sample is drawn*". Also, Saunders et al. (2012) affirmed sampling frame is "*a complete list of all the cases in the population from which your sample will be drawn*".

In pursuance of the above assertions by Saunders et al. (2012) and Sekaran and Bougie (2010), the sampling frame for the research constituted three Higher Education Institutions in Ghana (i.e. HEI-1, HEI-2, HEI-3) and offering different degree programmes in the field of computing, i.e., computer science, information technology, information systems and computer engineering. The assumptions for choosing the population from these institutions was that, these three HEIs are known to be technology driven universities that produce high quality graduates in Ghana in the area of computing sciences as an academic discipline. The instructional approaches employed among the participating universities also employed blended learning with their award-winning Learning Management Systems (LMS).

The researcher sampled the students from the targeted population and narrowed it down to only students and lecturers involved in the teaching and learning of programming courses at the undergraduate level. The programming courses targeted for the studies were Principles of Programming with C++, Introduction to Computing Science with C, C++ or Python, High Level Programming with C++, Java Programming, Advanced Programming with C#, Web Application Development with PhP, and Visual Basic (VB.net) programming languages.

4.10.4 Sample Size

Once an appropriate population and research sample frame were chosen, it was then imperative for the researcher to define the sample size for the study. Saunders et al. (2012); and Sekaran and Bougie (2010) asserted that defining a sample size requires consideration of several factors. Among these are:

1. Consideration of how to answer the research question;
2. Consideration of the margin of error and the level of data precision;
3. The time constrained and cost involved in the research; and
4. The type of data analysis to be conducted.

Studies by researchers such as Creswell and Poth (2018); Saunders et al. (2012); and Sekaran and Bougie (2010) affirmed that the higher the sample size, the lesser the level of margin of error and thus, increases the level of precision significantly. Saunders et al. (2012); and Sekaran and Bougie (2010) further asserted that when conducting a survey research, some researchers select sample sizes based on a portion of the entire population for instance, selecting 10% from the entire population or choosing the sample using previous experiences of other researchers' past results from their respective studies.

These approaches for selecting sample size have been refuted vehemently by a more recent study of Creswell and Poth (2018). Their main concern was that choosing such approaches will not contribute to gaining an optimal result for their research findings. They asserted that sample size should rather be selected based on the researcher's analysis plan. The researcher unequivocally supports the later assumption of Creswell and Poth on determining the analysis plan.

In view of the above assertions, the researcher employed the Slovin's statistical formula to calculate the sample size suitable for this study as indicated below:

$SS = N / (1+N * e^2)$

Where SS= Sample size;

N = the population of the study;

e = the confidence interval or margin of error (i.e. the margin of error used for the study was 0.05 representing 95% margin of error).

In computing the selected population and the value of the error of margin in the formula, the value of the sample size selected was 329. This value was defined to be the minimum sample size the researcher could choose per the targeted population for the study. Inferring from previous researchers, as indicated in the studies of Creswell and Porth (2018); Saunders et al. (2012); and Sekaran and Bougie (2010), that a higher sample size gives a lesser margin of error of margin and thereby increases the level of precision significantly. It was this motivation that the researcher believed that although the minimum sample size suitable for the researcher was 329 for the three selected HEIs, yet the researcher felt the sample size for the study considering the action research approach and iterative nature of the study should be 600. The rationale for choosing the 600-sample size was to establish a very strong argument for the appropriate techniques to be used for teaching computer programming in HEIs both in Ghana and globally.

In the light of this, a total of six hundred (600) students (see Table 4.4 for the distribution of the participants) from three different programming classes, in three different HEIs in Ghana constituted the sample size for the students. While a total number of 9 lecturers teaching programming at the undergraduate level at HEI-1, HEI-2 and HEI-3 constituted the targeted population for the teachers.

Table 4.4: Sampling Size

Institutions	**Population for Entire Computing Degrees**	**Selected Sample Size**	
	Population	Sample Size	Lecturers
HEI-1	660	200	3
HEI-2	620	200	3
HEI-3	600	200	3
Total	1,880	600	9

Source: Population and Sample of the Selected Universities, (2019)

4.10.5 Sample Technique

The sampling technique is the appropriate method employed for selecting the sample size, thus defining the selection process for the sample size (Creswell, 2018). The sampling technique that was adopted for the study was the non-probability sampling and thus employing the researcher's own judgements for collecting the data. According to Etikan, Musa and Alkassim (2016), a non-probability sampling technique is used when the defined sample does not include the totality of a given population. Consequently, the study employed the purposive sampling techniques for the study.

Etikan, Musa and Alkassim (2016) described purposive sampling as a sampling technique that allows the researcher to make judgements on the sampled data by deliberately choosing the respondents due to the respondents' qualities possessed. In other words, the researcher decided on the people who could cooperatively or willingly provide responses to the information needed for the research by virtue of their understanding or knowledge and expertise. Also, in order to enable the researcher to select the sample size on the basis of the nature of the degree programmes offered at the three universities, the researcher's personal judgements were used to select the students and the lecturers, thus the use of purposive sampling (Babbie and Mounton, 2011, p.166). The research sought to gather data from lecturers who the researcher thought could conveniently aid in employing the activity-based approach in the instruction of programming thus, purposively choosing them.

In the case of this study, the research focused on the development of the framework that could be used to teach programming effectively within a blended learning environment. This also accounted for the decision-making process requiring the judgement of the expert in the computer science education discipline.

4.11 Data Collection

The data collection phase of a research is crucial for making analysis and judgements (Cohen et al., 2012). In the light of this, the researcher administered two anonymous online-based questionnaires for the students and personally conducted a face-to-face open-ended interview for the lecturers respectively in semesters 1, 2 and 3 of the action research. The researcher also observed the teaching and learning approaches used for the instruction of computer programming. In some instances, the researcher also conducted focus group discussions of three different groups of students at the selected institutions of the study throughout the three semesters of the action research. Since the study employed an action research approach, the researcher also used his own personal experiences with his students to make assumptions for the study. Table 4.5 below depicts the research approach for collecting data based on the research question in the context of employing the mixed method approach. The main research question for the study was:

How should activity-based learning be incorporated into a blended learning environment in HEIs in Ghana?

Table 4.5: Research Questions and Instruments Adopted

Research Question	**Research Approach**			
	Literature Review	**Quantitative (online survey questions)**	**Qualitative (Interviews)**	**Qualitative (Focused Group Discussion)**
RQ1: What is the current activity-based learning approaches in computer programming instructions within a blended learning environment in HEIs of Ghana?	X	X	X	X
RQ2: What curriculum mediation based on pedagogic approaches could support activity-based learning for instructing computer programming in HEIs of Ghana?		X	X	X
RQ3: What is the learning and teaching gains of an activity-based learning approach in a blended learning environment for students and lecturers respectively among HEIs in Ghana?		X	X	X
RQ4: What new framework could be used to enhance activity-based learning approach in a blended learning environment for instructing computer programming in HEIs?	X	X	X	X

Source: Author's approach used for collecting data for each research question, (2019) The rationale for soliciting data for the research after gaining permission from the gatekeepers of the three selected universities in Ghana was based on the following premises;

1. To establish an affinity with the participants of the study and explaining details on the relevance of the study in his/her institution and the world in general.
2. To throw more light on aspect(s) where a participant needed clarification
3. To establish a greater percentage response rate for the participants for both the quantitative and qualitative data.

It was also important to do an empirical review of existing literature to gather data to build a new framework. The rationale for the empirical review was to establish existing variables that had been employed for the teaching and learning in HEIs that used blended learning.

The researcher conducted interviews with lectures and students at HEI-1, HEI-2, and HEI-3. The semi-structured interview formulation approach using open ended questions was used to capture data from both the students and lecturers, thus, to employ some level of flexibilities in the responses and gather the trends of feedback from the respondents. Appendix C depicts the interview guide for students on their experiences on the activity-based approach used during the first semester. Appendix G depicts the interview guide for the faculty on the developed ABLA framework for computer programming instruction in semester 3, Appendix D depicts the interview guide on the curriculum developed in semester 2, while Appendix F depicts the interview guide for the students on the ABLA framework in semester 3.

Focus group discussions among students were recorded and interpreted accordingly to gather data from a group or collective perspective experiences on activity-based learning approaches in programming instruction. Secondary sources of data were gathered from literatures of books, research journal articles, conference proceedings, theses or dissertations, websites, magazines and newspapers.

4.11.1 Questionnaire Administration

The researcher administered questionnaires to students and lecturers via online-based Google forms. This was accomplished after the researcher gained permission from the gatekeepers of the three participating universities in the research. Also, the participants were requested to read and acknowledge their consent to participate in the research before the questionnaire was administered to them. In all, a total of 600 students and 10 lecturers from all the three HEIs selected to participate in the research were directed to the link to complete the forms anonymously. In all, there were nine different sections of the questionnaire with different constructs. The rationale was to capture each variable for different constructs as indicated in the conceptual framework of the study. Copies of the questionnaire were administered to the students in the three HEIs from 2nd September 2019 to 30th October, 2019 for the 1st semester of 2019/2020 academic year via an online Google form. New data were collected from 1st February – 30th March, 2020 via an online Google form. Finally, qualitative data were collected from lecturers from the three HEIs from 1st April 2020 – 30th May, 2020.

Questionnaire Used for Semester 1 (Appendix B)

The sections of the questionnaire used for the first semester covered the following

Section A – Bio-data: This section captured the bio-data of the participants. E.g. Institution, gender, age, programmes of study, number of courses, level, etc.

Section B – Experiences of Activity-based Approach in Programming Instruction: This section also captured the:

- current approach for teaching computer programming,
- learners' experiences on the activity-based approach employed in their programming classes, and

Section C – Challenges of Learning Programming: this section captured:

- challenges faced in the teaching and learning of programming.

Questionnaire Used After Semester 2 (Appendix E)

The sections of the questionnaire used for the second semester covered the following:

Section A - Instructional Activities: This section captured the learners' experiences with their instructors for teaching programming using an activity-based approach within the blended learning environment. This denotes the teaching presence approach used in Garrison (2001, 2002) community of inquiry model.

Section B – Cognitive Activities: This section captured the learners' cognitive experiences of learning programming using an activity-based approach within the blended learning environment. This signifies the cognitive presence approach used in Garrison (2001, 2002) community of inquiry model and the constructivist model (Kanuka and Garrison, 2004; Wulf, 2005; Harasim and Harasim, 2018; Suhendi, 2018).

Section C - Social Activities: This section captured the learners' experiences with the social activities for teaching programming using the activity-based approach within the blended learning environment. This denoted the social presence approach used in Garrison (2001, 2002) community of inquiry model.

Section D- Learning Gains: This section captured the teaching and learning gains for employing the activity-based approach in teaching programming, thus establishing the personal fulfilment for the approach.

Section E – Critical Skills Development, Learning and Knowledge Attainment: This section established from the participants their experience on the activity-based approach and if it had contributed to their critical skills development towards their future career.

Section F – Feedback Activities: This section measured the feedback and assessment components of teaching and learning programming using the activity-based approach in a blended learning environment.

4.11.2 Interviews Administration

Apart from administering the research questionnaires, the researcher also conducted open ended interviews with nine students and nine lecturers who teach computer programming in the three participating universities for the research. The researcher gained permission from the gatekeepers of the three participating universities involved in the research and consequently requested the participants to read and acknowledge the consent note to participate in the research (See Appendix H). The interviews were conducted throughout the three semesters of the action research.

Interview Guide Used in Semester 1 (Appendix C)

The interview guide used in semester 1 established the students' learning experiences on the activity-based approach in their programming class.

Interview Guide Used in Semester 2 (Appendix D)

In Appendix D depicts the interview guide on the curriculum developed in semester 2, while

Interview Guide Used in Semester 3 (Appendix F)

Appendix F depicts the interview guide for the students on the ABLA framework in semester 3.

Interview Guide Used in Semester 3 (Appendix G)

Appendix G depicts the interview guide for the faculty on the developed ABLA framework for computer programming instruction in semester 3.

4.11.3 Focus Group Discussions

The researcher also took keen interest to conduct focus group discussions of three different groups of students consisting of three students in each group at the three different institutions totalling nine students in all. The rationale was to capture the students' opinions and personal experiences with the activity-based approach employed in their class within their blended environment. Similarly, three lecturers (one from each participating HEI) who teach programming were also engaged at the various participating institutions for the focus discussions on their teaching experience, pedagogic approaches employed and curriculum mediations issues. Before the focused group discussions were conducted, participants were requested to read and acknowledge the consent note to engage in the research. The focused group discussions for the students took place during the periods of 9th January – 3rd March 2020 before Government of Ghana's directives to HEIs to close down as a result of COVID-19.

Table 4.6: Demonstration of Analytical Methods and Data Analysis Used to Address the Research Questions

No.	Research Questions	Hypotheses of the Research	Data Source	Type of Data	Type of Analytical Technique used
1	What is the current activity-based learning approaches in computer programming instructions within a blended learning environment in HEIs.	H_1: The activity-based teaching and learning approach of computer programming in HEIs in Ghana within a blended learning environment include but not limited to case studies, quizzes, projects, group discussions and presentations, problem-based learning and concept mapping.	Primary data from online Questionnaire Interviews, focused group discussions	Nominal/ Ordinal for Quantitative (QUAN) and Text from respondents Qualitative (QUAL)	Descriptive analysis, 1-Way ANOVA Test, Multiple Regression Analysis/ Exploratory Factor Analysis, Confirmatory Factor Analysis, Thematic Analysis, Pattern Matching, and Logical Narrations.
2	What curriculum mediation based on pedagogic approaches could support activity-based learning for instructing computer programming in HEIs?	H_2: The new developed curriculum used in teaching computer programming within a blended learning environment supports activity-based learning.	Primary data from online Questionnaire Interviews, focused group discussions	Nominal/ Ordinal (QUAN) and Text from respondents (QUAL)	Action Research (Teaching, Observation and Reflection) Rapid Application Development (RAD) Thematic Analysis, Pattern Matching, and Logical Narrations.

3	What is the learning and teaching gains of an activity-based learning approach in a blended learning environment for students and lecturers in HEIs?	H_3: The activity-based learning approach of teaching computer programming within HEIs has a significant positive effect on students' learning gains. $H_{3.1}$: Th activity-based learning approach of teaching computer programming within HEIs has a significant positive effect on students' cognitive development. $H_{3.2}$: The activity-based learning approach of teaching computer programming within HEIs has a significant positive effect on students' social activities and learning.	Primary data from online Questionnaire Interviews, focused group discussions	Nominal/ Ordinal (QUAN) and Text from respondents (QUAL)	Descriptive analysis, 1-Way ANOVA Test, Multiple Regression Analysis/ Exploratory Factor Analysis, Confirmatory Factor Analysis, Thematic Analysis, Pattern Matching, and Logical Narrations.

		$H_{3.3}$: The activity-based learning approach of teaching computer programming within HEIs has a significant positive effect on students' teaching activities (Instructional activities). $H_{3.4}$: The activity-based learning approach of teaching computer programming within HEIs has a significant positive effect on students' engagement (Skills development). $H_{3.5}$: The activity-based learning approach of teaching computer programming within HEIs has a significant positive effect on students' feedback and assessments.			

		$H_{3.6}$: Social activities have a significant positive effect on learning gains.			
4	What new framework could be used to enhance activity-based learning approach in a blended learning environment for instructing computer programming?	H_4: The developed framework has a significant positive effect on activity-based learning approach in a blended learning environment for instructing computer programming.	Primary data from online Questionnaire Interviews, focused group discussions	Nominal/ Ordinal (QUAN) and Text from respondents (QUAL)	Structural Equation Model Confirmatory Factor Analysis, Thematic Analysis, Pattern Matching, and Logical Narrations.

4.12 Data Analysis

The data analysis stage of the study was prepared from the data derived from Stage 3 (i.e. Design of ABLA Framework) in the action research design process and interpreted the data for a larger understanding of the study (Creswell, 2018). Since this research employed the mixed method approach, the analysis of data was done both quantitatively and qualitatively.

Firstly, the quantitative data generated from the online survey was initially analysed and screened. An initial descriptive statistics and one-way ANOVA test were also conducted. The study used the Statistical Package for Social Sciences (SPSS), version 26, and AMOS 26 for the quantitative data analysis. The quantitative data was tested to check normality as to whether the data was appropriate for data analysis using Structural Equation Modelling (SEM) (Hair et al., 2014). Analysis on the normality test was conducted and all missing values were handled accordingly. After an analysis to measure outliers was performed on the data, the normality and the outlier checks were done; the researcher then employed Structural Equation Modelling to test the Hypotheses of the research that aided in answering the research questions. The structural model was then assessed using the Multiple Regression Analysis, path coefficients, the coefficient of determination R^2, predictive relevance Q^2 and the f^2 effect size. (Hair et al., 2014; Hair et al., 2011; Henseler et al., 2009, 2016). Confirmatory factor analysis (CFA) was then conducted to establish how the measured variables were represented on the constructs.

Secondly, the qualitative data analysis was conducted using Atlas Ti. V.8. The qualitative data were analysed using thematic content analysis, pattern matching, and logical narrations.

4.12.1 Structural Equation Modelling (SEM)

After the normality and the outlier checks were done, the researcher then used SEM to test the hypotheses of the research that could aid in answering the research questions. The structural model was then assessed using the Multiple Regression Analysis, path coefficients, the coefficient of determination R^2, predictive relevance Q^2 and the f^2 effect size (Hair et al., 2014; Hair et al., 2011; Henseler et al., 2009).

The reliability and validity of the variables were also tested to validate the measurement models. Also, the causal paths that existed between the variables as postulated within the conceptual framework were tested. Moreover, the analysis of the relative Importance-Performance Matrix analysis was done to establish an evaluation of relative importance performance variables that could aid in predicting other variables of the study.

Normality testing, factor analysis, inter-correlation analysis among sub factors and composite scores were employed to analyse the quantitative data. The structural equation models were used in order to aid in establishing the ABLA framework.

Analysis of variance was conducted to compare the students from the three different groups and semesters that used different ABLA design frameworks for teaching with the three different semesters of the case study research. The independent variable constituted the type of activity-based models and the dependent variable included the scores on the scale measuring learning achievements or gains, teaching activities, instructional activities, cognitive activities, student engagement, feedback and assessment.

In using the structural equation modelling and other test of the study, specific assumptions are required to be tested. As a result of establishing these assumptions, this section proposes appropriate assumptions need to use SEM. The assumptions made include: 'Multivariate Normality', 'Multicollinearity', 'Sample Size', 'Positive Definiteness' and 'Univariate Normality'.

4.12.2 Multivariate Normality

To test for multivariate normality in SEM, a linear regression needs to run with an ID as the independent variable and the other items as dependent variables. Afterwards, the Mahalanobis distance check was conducted to see if there were any outliers. One case fell below the expected probability level of .001 which was the maximum. Hence, that case was eliminated from further analysis. The Mahalanobis distance considers if there is an outlier after the aggregation of all the items for each case. To check for the Mardia's coefficient, according to Bentler (2005), "*in practice, values > 5.00 are indicative of data that are non-normally distributed. In this application, the z-statistic of 61.920 is highly suggestive of non-normality in the sample*" (as cited in Bryne, 2016, p.104). Appendix J depicts the assessment of normality and the full meaning of the coded variables.

4.12.3 Multicollinearity

In testing for multicollinearity, a regression output was tested. "*In the collinearity statistics under the coefficients table, the tolerance and Variance Inflation Factors (VIF) were screened for Figures <.01 and >10 respectively"* (as cited in Bryne, 2016, p.104). Since none of the tolerance figures was below .01 and the VIF above 10, the assumption of Multicollinearity was satisfied (Menard, 1995).

4.12.4 Sample Size

Measuring the suitability of the sample size for SEM, Soper, a web-based calculator was used (Soper, 2018). Upon performing the computation, the minimum sample size generated was 161. Hence, the 300 cases far exceeded the minimum required number of cases suitable for performing analysis using SEM.

4.12.5 Positive Definiteness

To determine that the assumption of positive definiteness was not violated, factor analysis was conducted. It is noted that in the correlation matrix table, "*the determinant value should not be equal to zero*" (as cited in Bryne, 2016, p.104). The observed determinant was not equal to zero i.e. (2.552^{-009}), therefore, the assumption of positive definiteness was not violated for the study.

4.12.6 Univariate Normality

The univariate normality was conducted using SPSS to measure the Shapiro Wilk test for significance. That is if the significance value is <.05, then, the null hypothesis is accepted and thus the data is significantly unusual when compared with a normal distribution. The variables in the study were all above the cut-off point of .05. The Shapiro-Wilk test results were .65, .71,

.55, .66, .65, .78, and .76 for SD, IA, FB, SA, AU, CA and LG respectively. Though the variables were not normally distributed, it is however not alarming because of the huge data involved (Pallant, 2016).

4.13 Reliability and Validity of the Research Data

Reliability, as used by Greener (2008, p.37), is a term to measure the consistency in the data and transparency in the research method which can provide confidence to the beneficiaries and readers of the research. In other words, the readers should be able to use the same approach to yield the same results in a research. Accounting for this, the researcher firstly conducted a test on the model and all the variables in the constructs to measure internal reliability and consistency of the data sets. As applied by most researchers, the Cronbach's α value (Cronbach, 1951) is used to determine the reliability of the variables. Although this approach is employed by most researchers, yet Hairet al. (2014) have raised critique of using the Cronbach α value for testing reliability. In the light of this, Hair et al. (2014) in a more recent research suggested that a composite reliability test ρ_A (Dijkstra and Henseler 2015b) should be used as an alternative to Cronbach α.

Hair et al. (2014) posit that the composite values within the ranges of 0.6 and 0.7 are acceptable for conducting exploratory research. Notwithstanding that, Hair et al. (2014) and Nunnally and Bernstein (1994) also asserted that an advanced research analysis accepts composite values within the ranges of 0.7 and 0.9. The rationale is that composite values of less than 0.6 depicts less reliability of the data. More so, composite values greater than 0.95 are also known to be detrimental to the reliability of the data because it might be possible that the test is measuring same construct (Hair et al., 2014).

In view of this, the researcher conducted the reliability test on the composite values and assessed the factor loadings of the individual indicators on the conceptual framework. The researcher deleted values that were not significant below or above the threshold, i.e. values between 0.4 and 0.7. Also, biases and errors during data capturing were reduced to a minimal level by enforcing an accurate data collection process and updated the data set where data was wrongly inputted.

The validity of the research findings or results are said to be valid when the research data is credible, thus demonstrating if the findings depict the realities about the research (Saunders *et al.,* 2007, p.150). By employing the SEM approach for analysing the data, the researcher employed two major validity tests on the data. These were the discriminant validity test and the convergent validity test.

The discriminant validity test is the test designed to measure the degree of the latent variables as to whether they are actually distinct from other latent constructs. This accounts for why the latent constructs should be uniquely defined in the model. The three major measurements for employing discriminant validity test were employed for the study.

Accordingly, the first step was for affirming that factor loadings of the latent constructs are higher than the cross loadings (Henseler, Hubona and Ray, 2016). Secondly, the Fornell-Larcker benchmark which confirms the Average Variance Extracted (AVE) of the latent construct was applied. The AVE is the variance measurement captured relative to the variances that occurred during error testing. The Fornell-Larcker criterion depicts the AVE scores of the latent construct which should be more than the square-correlations among the constructs (Henseler, Hubona and Ray, 2016). Lastly on the discriminant validity test, the Heterotrait-Monotrait Test (HTMT) was also performed to estimate the factor correlation (Henseler, Hubona and Ray, 2016). Henseler, Hubona and Ray (2016) asserted that to discriminate among factors, the HTMT should have a significant value less than one.

Similarly, the convergent validity test was also conducted to measure the magnitude to which an indicator absolutely correlates with other indicators of similar constructs positively. This on the other hand, promotes a convergent comparison of the constructs. The AVE values were used to measure the convergent validity of the constructs. A more recent studies of Henseler, Hubona and Ray (2016) averred that the AVE values for every construct should be more than 0.5.

Hence, the research employed different sources of verification to enhance the validity and reliability of the data collected from different sources using different methods which in effect helped to deepen and confirm the data.

4.14 Ethical Considerations

Ethical research concerns include quality data formulation, defining a research topic that is worth researching, quality review of related literature, the design approach, data collection approach, analysis of the data and formulation of the research findings devoid of academic dishonesty.
It is a requirement to carry out research in the University of South Africa devoid of unethical writing. Hence, UNISA requires a high standard of academic writings that conforms with the code of conducts of a professional academic writing devoid of plagiarism, ambiguities and uncertainties in the write-up etc. Also, proper concerns were required from UNISA to clear the researcher to conduct the research, i.e., the researcher gained an ethical clearance before conducting this research.

The researcher also received permission from the University of South Africa and the selected universities in Ghana to undertake the research. Hence, an ethical clearance was issued to the researcher with a reference number (029/EF/2019/CSET_SOC). See Appendix A for the ethical clearance certificate issued by the School of Computing Ethics Committee of UNISA.

It is important to establish that there exists guidance governing academic writing that needs to be followed, hence strict adherence of quality academic writing was followed. As much as possible, the researcher avoided voluntary distributing of academic writings that was not his own, causing harm to people, ambiguity, contravention of integrity, deceptive write-ups and breach of confidentiality which are highly unacceptable for scientific writings.

This research additionally sternly adopted good ethical behaviour that conformed to ethical standards free from falling victim to academic dishonesty or compromises in academic quality. More so, the findings of this study were honestly and sincerely presented.

The ethical issues below guided this research:

i. Data Integrity and Honesty: The researcher reported the research with sincere honesty and integrity with the data. That is, the data for the results and findings were true representation of the acquired data collected from the respondents, i.e. data from students, lecturers and administrators.

ii. Objectivity: The researcher performed the study devoid of biases in any component of the research such as the research design, analysis of the data, interpretation of the data and presentation of the research recommendations.
iii. Confidentiality: The researcher respected any form of data, interviews, or past record as highly confidential from all the population. The researcher also followed UNISA's code of conduct on confidentiality in terms of data gathering, hence ethical clearance was received before data was collected.
iv. Detailed Attention to Intellectual Property: The researcher also ensured that all forms of plagiarism and copying of other people's academic work without acknowledging appropriate references were not entertained in this study.
v. Protection of Humans: Since this research involved humans, the researcher made sure to reduce any forms of harm to a minimum level and rather increased its benefits to other people and the participants.

In a nutshell, the researcher ensured that the entire research was written free from plagiarism and thus, all resources and materials used were duly referenced. Brynard and Hanekom (2006) indicated that plagiarism occurs when a researcher uses someone's ideas, concepts or writings as though one is the legitimate owner of the source. In line with this, the researcher duly complied with the University of South Africa's ethical standards of doing doctoral research and fully acquired ethical clearance for this study.

4.15 Summary of Research Methodology Chapter

The methodology chapter of the study has shed light on the various approaches that aided the researcher to develop the framework for the implementation of activity-based teaching and learning in HEIs within a blended learning environment among undergraduate students. The researcher drew assumptions from existing theories and philosophical orientation, i.e. the constructivist worldview to guide the research. In the quest to develop the framework, the researcher adopted action research to implement the activity-based teaching and learning of computer programming within three higher education institutions in Ghana that use blended learning.

The researcher adopted the mixed method approach to capture both qualitative and quantitative data for the study within the action research design process. The action research design approach as designed by the researcher was the sequential procedure that the researcher followed to iteratively carry the research within three different semesters among the selected HEIs in Ghana. Consequently, the researcher selected the population, sampling frame and sample size for the study using statistical approach.

Additionally, the researcher described the data collection techniques and the data analysis aspect of the research. The researcher employed the structural equation modelling, inferential statistics, and thematic analysis to successfully analyse the data to answer the research questions. Therefore, the reliability and validity test of the data was deeply verified vis-à-vis their Cronbach alpha values. The SPSS, SPSS AMOS, and Atlas Ti. were the main tools employed for analysing the data for the study.

Finally, the researcher took a keen attention to obtain an ethical clearance of the research in order to ensure confidentiality, integrity, anonymity and consideration of data privacy and protection of participants against all forms of harm during data elicitation stage of the research.

CHAPTER 5

PRESENTATION OF RESULTS AND ANALYSIS

5.1 Introduction

In the previous chapter, the researcher outlined the methodology used for conducting this research. The methodology was deployed and the data was captured using a statistical package. The main aim of this study was to develop a framework for activity-based programming instruction within a blended learning environment of Higher Education Institutions (HEIs). Data captured in this study has been analysed and the results obtained from the data collected have been presented in this chapter for the first and second semester of the action research.

The analysis in this chapter is organised per the action research based on the data received from the mixed method approach from the students and lecturers. The SPSS 26 software was used for the preliminary analysis, ANOVA test on the quantitative data, and Atlas Ti software was used for the qualitative data analysis respectively.

5.2 Analysis of First Semester Results of the Action Research

The first data collection was conducted in the first semester of the action research year which aimed to establish the demographic characteristics of the students and lecturers, the current activity-based teaching and learning approaches and the challenges in teaching and learning of computer programming among the three HEIs. See Appendix B for the questionnaire used to gather the data for the first semester.

5.2.1 Response Rate

The study targeted 600 respondents to the questionnaire among the three HEIs initially using purposive sampling technique (Creswell, 2018). However, the researcher received 300 completely filled questionnaires representing 50% success rate towards the targeted sample. Consequently, Baruch and Holtom (2008:1139) averred that the 50% response rate is seen to be satisfactory. Therefore, the 300 responses received were used for this study. The data was further screened to ascertain the usability, reliability and validity of the data before an in-depth analysis were conducted on the data. There were no missing values, i.e. all respondents fully completed their questionnaire as indicated in Table 5.1.

Table 5.1: Response Rate of the Sample

Item	Number
Questionnaire Administered	600
Initial total responses	300
Non-usable responses	0
Total usable responses	300
Usable response rate	50%

Baruch and Holtom (2008:1139) reviewed 1607 published papers in the years 2000 and 2005 and assessed 17 refereed academic articles and did a meta-analysis of 490 articles and found an average response rate of 52.7% with a standard deviation (sd.) of 3.4 for data collected from individuals. However, the mean response rate for studies within institutions and organisations was 35.7% with sd. of 18.8. Therefore, the response rate (50%) used for this study for three institutions was seen as acceptable per the studies of Baruch and Holtom (2008, p.1139).

5.2.2 Reliability Analysis (Data for Semester 1)

A reliability test was performed on the students' responses to the questionnaire to ascertain whether the data received in the first semester were reliable or not. Table 5.2 below depicts the reliability results of the data collected from the students in the first semester. The reliability analyses determine the scale's internal consistency (Salkind, 2018). One of the most commonly used indicators of internal consistency is the Cronbach's alpha coefficient (α). Ideally, the Cronbach alpha coefficient of a scale should be above 0.7 (Salkind, 2018).

Table 5.2: Reliability Analysis

	No. of Items	Cronbach's Alpha	Acceptable Level
Total Items	43	0.866	Good

The outcome of the reliability test conducted attested that all the defined constructs had a Cronbach's alpha coefficient (α) more than 0.7, i.e. α=0.866.

5.2.3 Demographics of Respondents

Respondents for this research were made up of students from three technology-based universities in Ghana. As mentioned earlier, 300 respondents completely filled the questionnaire which were usable for this study. The students were asked to indicate their demographic information such as gender, age, institutions, academic program, enrolled level, etc. Table 5.3 summarises the demographic characteristics of the respondents.

Table 5.3 Characteristics of Respondents

Variable	Category	Frequency	%
Gender	Male	271	90.3
	Female	29	9.7
	Total	300	100
Institutions	HEI-1	152	50.7
	HEI-2	103	34.3
	HEI-3	45	15.0
	Total	300	100
Program (Bachelor / Diploma)	Information Technology	278	92.7
	Computer Science	16	5.3
	Computer Engineering	3	1.0
	Information Systems	3	1.0
	Total	300	100
Age	18-20	91	30.3
	21-24	99	29.3
	25-30	78	26.0
	36 Above	32	10.7
	Total	300	100
Level	100	164	54.7
	200	11	3.7
	300	38	12.7
	400	87	29.0
	Total	300	100

Among the 300 students who responded to the survey, 271 (90.33%) were recognized as male while 29 (9.67%) identified themselves as female. The gender imbalance in the fields of computing and sciences is known to favour male students over females. Female students are not mostly seen in the domain of sciences as compared to social sciences and humanities in most HEIs globally (Huyer, (2015). With respect to age, the majority of the students were within the ages of 18 and 20 (30.33%) and 21 and 24 (29.33%) respectively. This reflects most young students reading undergraduate programmes at the three HEIs in Ghana. Only a few students were above the age of 30, representing 14.33%.

The outcome of the analysis revealed that the majority of students (152; 50.67%) who responded to the survey were from HEI-1 whilst 103 (34.33%) of the students were from HEI-2. The high response rate from these two HEIs was as result of the researcher's influence on the faculty and students. Regarding students' levels on their academic programmes among the three HEIs, 164 (54.67%) were identified as level 100 students. This is quite understandable since most of the foundational programming courses are taught at levels 100 and 200. However, advanced programming courses are taught at levels 300 and 400.

5.2.4 Computer Programming Courses Enrolled by the Respondents

The outcome of the analysis revealed that 55.3% of the respondents had enrolled in 1 to 2 courses, and a good number (41.7%) had enrolled on 6 or more courses which is a standard practice of most semester courses as indicated in Table 5.4. Additionally, the majority of the respondents (72.3% and 12.3%) had enrolled in 1 or 2 programming courses respectively. It was also noted that few students (3.7% and 11.7%) had enrolled in 3 or 4 programming courses. Of course, most computer science programs blend other elective courses with the programming courses.

On the programming languages enrolled by the students, the programming language of the respondents were largely C++, Vb.net, C# and PhP. Other programming languages such as Python, C, and ASP.net were not largely enrolled by most students in the three HEIs.

Table 5.4 Computer Programming Demographics of Respondents

Variable	Category	Frequency	%
Courses Enrolled	1-2	166	55.3%
	3-5	9	3.0%
	6 or More	125	41.7%
	Total	300	100%
Number of Programming Courses Enrolled	1	217	72.3%
	2	37	12.3%
	3	11	3.7%
	4 or More	35	11.7%
	Total	300	100.0%
Programming Language Enrolled 1	C++	82	27.3%
	Java	48	16.0%
	PhP	12	4.0%
	C	1	0.3%
	Python	2	0.7%
	C#	68	22.7%
	Other	87	29.0%
	Total	300	100.0%
Programming Language Enrolled 2	Nil	269	89.7%
	Java	16	5.3%
	C	1	0.3%
	C#	3	1.0%
	ASP.Net	4	1.3%
	Other	5	1.7%
	Total	300	100.0%
Programming Language Enrolled 3	VB.Net	292	97.3%
	PhP	2	0.7%
	C	1	0.3%
	Python	2	0.7%
	Other	3	0.7%
	Total	300	100.0%

5.2.5 Characteristics of Respondents' Computing Literacy

The study also enquired about the respondents' computing literacy levels to establish their levels with the use of computer. As depicted in Table 5.5 it was noted that 21.7% of the respondents were beginners with the use of computer, 68.3% were intermediate on computer proficiency and 10% on advanced level of computing usage. The internet experience of the respondents was also asked. 7% of the respondents had 2 years of internet experience, 7% 3 years’ experience, 0.7% had 4 years’ experience, 7.7% had 5 years’ experience and the majority (77.7%) had more than 5 years’ experience.

Table 5.5 Characteristics of Respondents’ Computing Literacy

Variable	Category	Frequency	Percent %
Computer Proficiency	Beginner	65	21.7%
	Intermediate	205	68.3%
	Advanced	30	10.0%
	Total	300	100%
Internet Experience	2 years	21	7.0%
	3 years	21	7.0%
	4 years	2	0.7%
	5 years	23	7.7%
	More than 5 years	233	77.7%
	Total	300	100%

5.2.6 Analysis of the Current Activity-based Teaching and Learning Approaches Used in Computer Programming Instruction (First Semester)

To establish the current activity-based learning approach in computer programming instruction within the three HEIs, the students were requested to indicate the activity-based approaches used. The mean responses and the standard deviation of the responses were analysed and interpreted as indicated in Table 5.6. Findings from the studies indicated that the activity-based approaches that are used include group discussions and presentations with mean score of 3.64 and standard deviation of 1.039, quizzes with a mean of 3.58 and projects with mean scores of 3.27.

Based on the outcome of the analysis, it could be established from the data that these approaches support the teaching and learning of computer programming. However, it could also be deduced that the core of comprehension and coding of programming concepts to solve real-world situation while using problem-solving approaches (Dewey, 1938, Piaget, 1950 Bruner 1996, Ben-Ari, 1998, Grover and Pea 2013) were occasionally practiced.

Table 5.6: Activity-based Approaches Employed

Activity-based Approaches	Mean	Standard Deviation	Interpretation
Case Studies	2.80	1.090	Occasionally Practiced
Quizzes	3.58	0.945	Frequently Practiced
Projects	3.27	1.307	Frequently Practiced
Group Discussions and Presentations	3.64	1.039	Frequently Practiced
Educational Games	2.01	1.126	Rarely Practiced
Videos	2.86	1.361	Occasionally Practiced
Debates	2.12	1.156	Rarely Practiced
Problem-based Learning	3.01	1.270	Occasionally Practiced
Round Table Discussions	2.49	1.132	Rarely Practiced
Peer Review	2.60	1.227	Rarely Practiced
Field Work	2.08	1.174	Rarely Practiced
Concept Mapping	2.08	1.180	Rarely Practiced
TEL Tools	2.75	1.228	Occasionally Practiced

The occasional practice of a problem-solving approach is contradictory to the effective teaching and learning approaches in computer programming where students are supposed to construct their knowledge to solve real world problems. Ben-Ari (1998) asserted that problem-solving and critical thinking approaches enhance an active engagement of students and consequently influences their programming skills.

Furthermore, it was deduced from Table 5.6 that the use of videos and Technology Enhanced Learning (TEL) tools such as Socrative, Padlets, Google Docs, etc., was noted to be occasionally practiced in the teaching and learning of computer programming. Again, these useful tools were relevant to enhance the comprehension of programming. However, it was least to be desired with a mean of 2.86 and 2.75, respectively. In the studies of Walker, Voce and Ahmed (2012), they posit that using a TEL approach in learning supports active engagement by students and also enhances their performance and learning experience. Moreover, the use of educational games to enhance programming debates, round table discussions of critical issues, peer review, fieldwork and concept mapping were rarely practiced in the teaching and learning of computer programming among the three institutions.

5.2.7 One-Way ANOVA Test of the Activity-based Approaches Used

A one-way ANOVA test was conducted to examine if there were any statistical difference among the three institutions in terms of the activity-based instructional approaches in teaching computer programming. It was noted that there existed a statistically significant difference of ($p=0.00$) among the three institutions as indicated in Table 5.7. If ($p < 0.05$), then it is established that there is statistically significant difference between the three groups (Creswell, 2018).

Table 5.7 One-Way ANOVA Test of the Activity-based Approaches Used

	Sum of Squares	**df**	**Mean Square**	**F**	**Sig.**
Between Groups	48.427	2	24.214	25.298	.000
Within Groups	284.267	297	.957		
Total	332.694	299			

The outcome of the one-way ANOVA test re-affirms the fact that there were different lecturers with different professional experiences, different teaching styles, different learning environments, and also different orientations of knowledge, etc. The difference among the three HEIs is practically understandable. However, the differences in the activity-based approaches must be tailored towards enhancing students' learning experiences and performances in computer programming, but the findings from the study revealed an abysmal indication with respect to its effective usage.

5.2.8 Challenges of Learning Programming

After ascertaining the various activity-based approaches used in the instruction of computer programming, the researcher also examined the challenges faced by students in computer programming classes (see Part C of Appendix B for the instruments used). The rationale for ascertaining the prevailing challenges faced by the students were issues including students' understanding of the syntaxes and semantics of programming constructs. The students were requested to indicate their responses from a 5-point Likert scale from 1-5, i.e. (Strongly Disagree, Disagree, Neutral, Agree and Strongly Agree). The challenges, as indicated in Table 5.8, were followed by a worrying trend that calls for a critical re-engineering of the pedagogic approaches in teaching computer programming in higher education, especially among the three HEIs in Ghana. The mean responses and the standard deviation of the responses were analysed and deduced.

Table 5.8 Challenges of Learning Programming

Challenges	**Mean**	**Standard Deviation**	**Interpretation**
Using programming development environment	3.00	1.060	Neutral Responses
Gaining access to computers/Internet	3.00	1.255	Neutral Responses
Understanding programming structures	3.36	1.010	Most Students Agree
Learning the programming language syntax	3.31	1.079	Most Students Agree
Designing a program to solve a certain task	3.37	1.021	Most Students Agree
Dividing functionality into procedures	3.37	0.999	Most Students Agree
Finding bugs from own program	3.30	1.089	Most Students Agree

The results on the challenges faced by the students from the study indicated that the use of programming development environments, gaining access to computers and internet connectivity were not noticed as a challenge since most students responded neutrally with a mean of 3.00 and standard deviations of 1.06 and 1.255, respectively. This translates to an affirmation that the three HEIs are technology driven institutions and thus gaining access to basic programming development environments tools, computer labs and internet was predominantly not an issue among students. However, most of the students agreed to the fact that understanding programming structures and semantics are very difficult to comprehend with a mean response of 3.36 and standard deviation of 1.010.

5.2.9 One-Way ANOVA Test on the Challenges of Learning Programming

Intriguingly, the findings from the three HEIs were not different from each other with respect to the preponderant challenges faced among the students in programming classes. To establish this notion, a one-way ANOVA test was conducted to examine if there were any statistical differences among the three institutions in terms of the challenges faced in learning computer programming.

Table 5.9: One-Way ANOVA Test on the Challenges of Learning Programming Instructions

	Sum of Squares	**df**	**Mean Square**	**F**	**Sig.**
Between Groups	1.589	3	.530	.811	.489
Within Groups	193.378	296	.653		
Total	194.966	299			

It was noted that there was no statistically significant difference between the three HEIs since the p-value was greater than 0.05. The p-value was 0.489 among the three institutions as indicated in Table 5.9.

5.2.10 Findings from the Qualitative Analysis of the Study for the First Semester

The second aspect of the action research following the mixed-method approach for the first semester constitutes the use of qualitative data for the study (see Appendix C for the instruments used to collect the qualitative data for the first semester). It is in this regard that the researcher conducted semi-structured interviews and focused group discussions to ascertain the views of the students and faculty on the current activity-based approach used in teaching computer programming within a blended learning environment.

5.2.10.1 Profile of the Participants

A total of six students responded to the qualitative data gathering instrument via a self-administered interview and focused group discussions. This was done in a face-to-face meeting before COVID-19 from 30th September, 2019 – 30th October, 2019. The demographics of the students who responded to the qualitative interview are demonstrated in Table 5.10. For confidentiality purposes, the interviewees are represented as Student A, Student B, Student C with their Institutions as HEI-1, HEI-2 and HEI-3. The interviewees were predominately Information Technology and Computer Science students. With gender, there was a gender balance with respect to each institution. Predominately, the levels of students interviewed were in levels 100, 200, and 300.

Table 5.10: Demographics of Interviewees

Interviewee	Gender	Institution	Level	Program
Student A HEI-1	Female	HEI-1	100	Information Technology
Student B HEI-1	Male	HEI-1	300	Computer Science
Student C HEI-1	Male	HEI-1	200	Computer Science
Student A HEI-2	Male	HEI-2	100	Information Technology
Student B HEI-2	Female	HEI-2	200	Information Technology
Student C HEI-2	Male	HEI-2	300	Computer Science
Total Responses = 6				

5.2.10.2 The Students' Learning Experience and Perception on the Current Activity-Based Approach Used in Computer Programming Instruction

Firstly, the students were asked to provide their learning experiences and perceptions on the activity-based approach used in teaching computer programming. The researcher wanted to find out how most students perceive the activity-based approach and consequently how the approach affects their learning journey in computer programming classes.

The findings from the study indicated that most students are appreciative towards the use of the activity-based learning approach in teaching and learning computer programming. However, some students expressed interest for improvement and more engagements of computer programming assignments and activities. Most of the students indicated that the activity-based approach used in their current semester included quizzes, project works, case studies, group presentations, and the use of video tutorials.

The comments raised by the some of the interviewees in the focused group discussions are indicated below:

Interviewee Student A HEI-1
"The activity-based approach employed in the programming classes are effective but need more improvements. The group projects, hands-on projects in class and the video tutorials were very useful to me".

Interviewee Student B HEI-2
"The activity-based approach is the best as far as teaching a practical course like programming is concerned, however, programming involves more practical than the usual theory. So, I appreciate my lecturers' approach of using the activity-based approach to teach especially with the hands-on activities in the classroom when problems are given to me to solve".

Interviewee Student C HEI-3
"It's very helpful. Once involved, you never forget. Oh! Yes, the activity-based approach as employed in teaching and learning in my school helped to improve my understanding of concepts taught. It also aids my retention as a computer science student".

Interviewee Student B HEI-1
"The activity-based approach makes learning fun. Though I make mistakes yet I discover solutions. It is a good practice and it helps me to deduce solutions to problems myself. It has really enhanced my learning experience".

Interviewee Student C HEI-2

"It is a good way to facilitate learning. Its helps people to express out their various problems in information technology so they can be tackled in class. Also, there should be more practical work".

Interviewee Student C HEI-3

"I could not program most of the assignments in the classroom or even assignments. I mostly get external help from colleagues and my seniors before I can submit my assignments".

It was noted from the students' comments that most of them expressed diverse opinions to the responses on their experiences of the current activity-based approach used in their respective institution. It was established that the approach was helpful to some of the HEIs, example is in the case of Interviewee Student A HEI-1, Student B HEI-2, and Student C HEI-2, however, some students see programming as very challenging as Student C HEI-3 asserted.

Interestingly, findings from the interview responses showed that most students have the desire to learn computer programming, however, they faced difficulties in understanding the syntax and logic of the programming languages. The comments from the interviewees are indicated below:

Interviewee Student A HEI-3

"I have interest in programming even before applying to my university but the programming lecturers don't teach well for you to understand the concept and that makes it difficult for you to program."

Interviewee Student A HEI-1

"The programming sessions need a peer group reading, projects, and assignments. In each activity was very good and well-paced. This approach as employed by lecturer has actually sharpen my brain in solving real world problem."

Interviewee Student A HEI-2

"The class was very involving and most often pair programming, case studies, class discussions, group presentations contributed to my understanding to program."

Interviewee Student B HEI-2

"I don't like programming and the school doesn't even stress on it to be done well. Understanding the syntax and semantics has always been a war for me."

Interviewee Student B HEI-1

"The feedback from my peers was very effective and aids my skills in programming".

Interviewee Student B HEI-3

"It was difficult at first but as time goes on I understood slowly. I love to program always".

Interviewee Student C HEI-1

"The lecturer doesn't teach to my understanding He just comes to class and projects codes for us to enter on our laptops without explaining why we use identifiers, variables, arrays, etc."

Interviewee Student C HEI-2

"It should be practically taught, the very basic of programming is never taught by my lecturer. He always gives too much assignments and activities. Meanwhile, there are other courses unattended to, my interest is not in programming".

Interviewee Student C HEI-3

"Overall, I think the method adopted by the lecturer has improved my studies".

Interviewee Student C HEI-1

"Again, in my view I believe necessary materials for the course has to be provided to make it interesting. Secondly, I think the lectures should make it their core mandate to let the students understand the concept they teach."

Clearly, it is evident from the responses from the interviewee students that there are issues with teaching and learning of computer programming. Upon a critical analysis of the responses from the students, it could be deduced that most of the students lack the interest to learn computer programming, and also lack understanding of the syntax and semantics and thus see the course as difficult. Upon further probing, most students responded in the affirmative that they "don't hate the course," however, according to them, the approach some of their lecturers' use is not helping them especially as commented by Interviewee Student C HEI-1, Interviewee Student C HEI-2, Interviewee Student B HEI-2, and Interviewee Student A HEI-3.

It was, however, contended by some of the students that they have the desire to learn computer programming and thus see programming as a fun. The researcher believes that intrinsic factors such as self-confidence, self-motivation, level of satisfaction and comfort (Smith and Ungar, 1995; Robins et al., 2003) are key factors for learning programming and thus saw from the comments from the students who averred positive desire to programme as those who were intrinsically motivated as commented by Interviewee Student A HEI-1 and Interviewee Student A HEI-2. It was also interesting to note that some students believe that programming in pairs, groups and taking feedback from their peers really contributed a significant interest in programming as commented by Interviewee Student A from HEI-2.

5.2.10.3 Relevance of the First Semester Findings on the ABLA Framework Development

The analysis of data received from the first semester has shed light on the demographic characteristics of the students and lecturers, the current activity-based teaching and learning approaches and the students' learning experience and the prevailing challenges in teaching and learning of computer programming among the three HEIs in Ghana. The findings established in the first semester aided the researcher to confirm the characteristics of the respondents, the population and sample of the research and their perception on computer programming instruction within a BL environment. The findings contributed about 30% of the action research towards the development of the ABLA framework. This is represented in Figure 5.1 below.

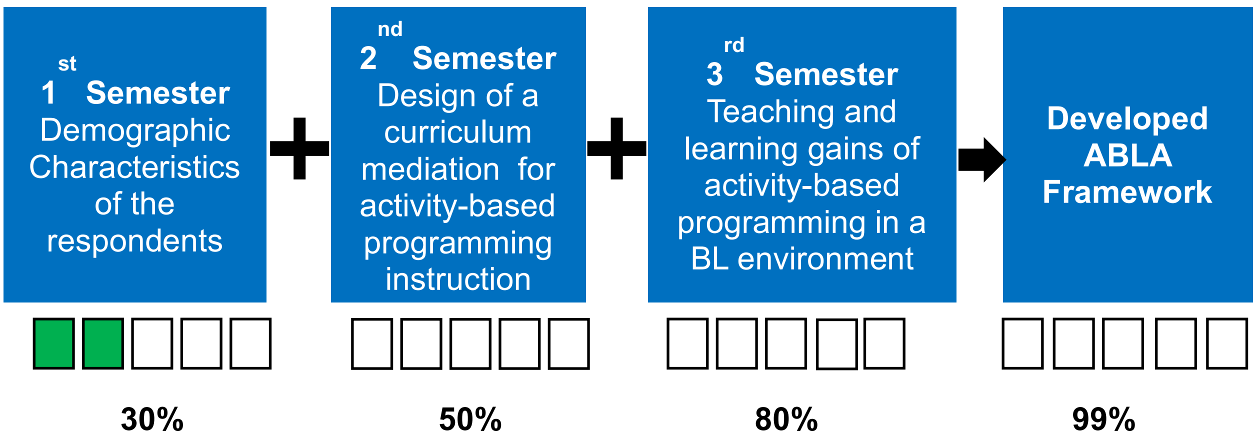


Figure 5.1 Progress View Towards ABLA Framework Development as at Semester one

In the subsequent sections, the researcher presents the analysis of the data received during the second semester activities of the AR towards the development of a curriculum mediation for computer programming instruction within a BL environment.

5.3 Analysis of Second Semester Activities of the Action Research

At the beginning of the second semester of the action research, the researcher shared the analysed data derived from the students during the first semester with the lecturers. The data helped the researcher and the instructors in the three HEIs to evaluate the activity-based approach they have being using, the student's learning experiences of programming instruction and the prevailing challenges the students faced.

Afterwards, the researcher developed a curriculum mediation for activity-based computer programming instruction within a blended learning environment for a standard 14-week academic period. Qualitative data from six computer programming instructors were collected to analyse and evaluate the designed curriculum. A web-based computer programming instructional platform was likewise developed and implemented to validate the relevance of the developed curriculum amidst COVID-19 in teaching computer programming either synchronously or asynchronously. Further, the section discusses the instructor's experiences and reflections on the activity-based curriculum developed for the instruction of computer programming during the second semester of the action research periods. The user interface of the system and how the system operates has been captured in this section.

5.3.1 Design of Curriculum Mediation Based on Pedagogic Approaches that Supports Activity-Based Learning for Instructing Computer Programming in HEIs.

To measure the viability and effective use of the activity-based approach in computer programming instruction, the researcher developed, tested and deployed an activity-based programming curriculum. Following a reflection on the RQ2 and upon completing the first semester of the action research, the researcher developed a curriculum that was adapted among instructors teaching programming at the researcher's university (HEI-1), and also shared and used among other instructors who teach the same courses at the other universities, i.e. HEI-2 and HEI-3, all based in Accra, Ghana.

The researcher together with the other instructors agreed to use the curriculum or syllabus developed by the instructor for teaching their assigned programming courses that is C++ Programming, Introduction to Computer Programming, High-Level Programming I, and Java Programming. The researcher adapted a curriculum used by the HEI-1's UK programmes and approved by the National Accreditation Board of Ghana. The adapted curriculum was somehow relevant; however, it had pedagogic gaps in teaching programming using an activity-based approach that were addressed in the developed curriculum as seen in the next section.

5.3.2 Developed Curriculum for Computer Programming

The course code and name or title of the programming courses vary among the three universities but the instructors, upon several deliberations and consultations, agreed to the underneath (Table 5.11) curriculum or course outline or syllabus as indicated in this section.

Table 5.11: Course Codes and Titles

Institutions	Course Code	Course Title	Credit Hours
HEI-1	GTU121COM CS101 / IT101	Introduction to Computing (UK Programmes) Principles of Programming	20 Credit Accumulation and Transfer Scheme (CATS) points 3 hrs / week
HEI-2	CS102	Programming I	3 hrs / week
HEI-3	CS101	Programming I	3 hrs / week

It was noted among the three HEIs that the course titles were very similar to each other, however, the course codes differ among them. In all, the three HEIs run programming courses as foundational courses in Computer Science and Information Technology degree and diploma programmes. The credit hours used for synchronous and asynchronous delivery per week were 3 hours while the HEI-1 UK programmes use the Credit Accumulation and Transfer Scheme (CATS) of 20. All the programming courses running among the 3 HEIs had no pre-requisite course, since they were all foundational courses.

5.3.3 Aims and Summary of the Course

The course introduces the fundamental concepts of programming that underpin the technical and theoretical content of undergraduate degree courses based on the discipline of Computing. Students taking the module will develop core skills in programming by learning and applying syntax, problem-solving strategies and theory common to most programming languages. At the same time, professional practices associated with the industry will be covered, which include key software development concepts such as testing code, version control, functional decomposition; and interacting with non-technical colleagues and clients.

5.3.4 Intended Module / Course Learning Outcomes

The intended learning outcomes envisaged that on completion of the course or module, the student should be able to:

1. Demonstrate the ability to use basic control flow syntax to produce working solutions to problems.
2. Reason about simple algorithms, selecting or creating algorithms to solve specific and generalized problems and expressing them in a suitable manner.
3. Evaluate their work in a useful manner, using technical skills such as debugging and code testing; as well as reflective practices on their academic activity.
4. Employ procedural and O-O approaches to design efficient algorithms that use appropriate data structures.
5. Select and exploit appropriate features of the C++ programming language to implement solutions.

5.3.5 Indicative Contents

The indicative contents that were taught include:

1. Programming practice: selection, iteration, functions, recursion, data types, data structures, classes and objects.
2. Problem solving: methods of analysing a problem, functional decomposition and recursive algorithms.

3. Programming theory: Boolean logic, the concepts of complexity of algorithms and the computability of problems, distinguishing programming languages by their type systems and the programming paradigms supported.
4. Procedural programming
5. Top-down problem solving
6. Pseudo-code
7. Introduction to C++ Development
   - Variables and data types
   - Input and Output
   - Sequence, selection and iteration
   - Functions
   - Pointers
8. Sorting and searching algorithms
9. Data structures
   - Arrays
   - Stacks
   - Queues
   - Heaps
   - Trees

5.3.6 Methods of Assessment

The assessment methods followed both formative and summative approaches (Chew, Jones and Turner, 2008; Olsson and Mozelius, 2016). Formative assessment was seen as useful to prepare students for summative assessment and gives students an early indication of their progress towards the course intended learning outcomes. Table 5.12 below demonstrates the assessment methods used for teaching computer programming.

The assessment practices were tailored following the best practices and standards as detailed in the studies of Carless et al. (2011); Fluckiger et al. (2010); Gilbert et al. (2011); Miller et al. (2010). The composition of entire course delivery was capped at 100%. The breakdown of the assessment gradings were however different among the three institutions. As regards the HEI-1 UK programmes, 60% Coursework and 40% Exams are required. On the other hand, HEI-1 main programmes grades were 30% course work and 70% for examination marks. The situation in HEI-2 and HEI-3 was seen to follow the same trend of 40% for course work and 60% examination marks.

Table 5.12 Assessment Methods

Assessment		**Component Weighting**	**Learning Outcomes**				
			1	2	3	4	5
Coursework (Formative)	Quizzes Projects Assignments Phase Test 1 - 3	HEI-1 UK (60%) HEI-1 (30%) HEI-2 (40%) HEI-3 (40%)	Y	Y	Y		
	Semester Project Group Presentations Phase Test 4 - 6					Y	Y
Exam (Summative)	2-hour Exam or 3-hour Exam	HEI-1 UK (40%) HEI-1 (70%) HEI-2 and HEI-3 (60%)	Y	Y	Y	Y	Y
Total		100 %					

The assessment methods for the formative assessment was expected to achieve at least learning outcomes 1, 2, and 3 for the first phase and subsequent phases for learning outcomes 4 and 5. Notwithstanding the formative assessments, the summative assessment was expected to cover all the learning outcomes.

Generally, to pass an assessment for any programming course among the three HEIs, it is required of a student to at least pass both the course work and the final examination. The pre-requisite to pass the coursework must be at least 35% and examination must be at least 35% and the entire course or module mark must be at least 40% for HEI-1 and HEI-2. However, HEI-3 requires an average pass mark of 50% for both course work and examination. In the event of a student failing a course, that student is required to take the course as a new course or re-sit the assessments.

5.3.7 Suggested Activity-Based Pedagogic Approaches

The learning outcomes were carefully analysed by the instructors and consequently, the researcher provided a suggested activity-based approach that aids the teaching and learning of programming. Instructors were not restrained from using any specific activity-based pedagogic approach in delivery, however suggested activity-based approaches such as quizzes, case studies, problem-solving, group discussions, presentations, use of Technology Enhanced Learning tools like Socrative, Padlets and Screen Cast O-matic were highly recommended for teaching and learning.

Moreover, each activity-based approach employed comes with its own responsibilities for either the student or the instructor. In most cases, it was suggested that an instructor must do the activities while the students follow the instructions from the teacher as demonstrated in Table 5.13.

Table 5.13 Suggested Activity-Based Pedagogic Approaches

Learning Outcomes	Suggested Activity-based Pedagogic Approach (Not limited to any innovated approach)	Duration / Contact Hours	Responsibility (Not limited to any innovative approach)
Demonstrate the ability to use basic control flow syntax to produce working solutions to problems.	Oral Presentation Quizzes Case Studies Problem-solving Group Discussions and Presentation	Two lecture periods (6 hours)	• Teacher writes code in the classroom while the students follow. • Student writes code (outside classroom). • Teacher to guide students while students write code in the classroom.

			• Teacher guides students while the students solve the problem in groups.
Reason about simple algorithms, selecting or creating algorithms to solve specific and generalized problems, and expressing them in a suitable manner.	Problem-solving Quizzes Case Studies Group Discussions Projects Games Field Trips Video	Three Lecture Periods (9 hours)	• Teacher writes code in the classroom while the students follow. • Students write code (outside classroom). • Teacher to guide students while students write code in the classroom. • Teacher guides students while students solve problems in groups. • Teacher to use Technology Enhanced Learning (TEL) Tools (e.g. Padlet, Socrative). • Teacher to flip teaching (i.e. teacher to share video resources to students to watch via the LMS and report while the teacher leads the discussions).
Evaluate their work in a useful manner, using technical skills such as debugging and code testing; as well as reflective practices on their academic activity.	Problem-solving Quizzes Case Studies Presentations Group Discussions Projects Educational Games Individual reflections	Three Lecture Periods (9 hours)	• Teacher writes code in the classroom while the students follow. • Students write code (outside classroom). • Teacher to guide students while students write code in the classroom. • Teacher guides students while students solve problems in groups. • Teacher to use Technology Enhanced Learning (TEL) Tools (e.g. Padlet, Socrative). • Teacher to flip teaching (i.e. teacher to share video resources to students to watch via the LMS and report while the teacher leads the discussions).
Employ a procedural and O-O approaches to design efficient algorithms that use appropriate data structures.	Problem-solving Quizzes Case Studies Presentations Group Discussions Projects	Three Lecture Periods (9 hours)	• Teacher writes code in the classroom while the students follow. • Students writes code (outside classroom). • Teacher to guide students while students write code in the classroom.

			• Teacher guides students while students solve problems in groups. • Teacher to use Technology Enhanced Learning (TEL) Tools (e.g. Padlet, Socrative). • Teacher to flip teaching (i.e. teacher to share video resources to students to watch via the LMS and report while the teacher leads the discussions).
Select and exploit appropriate features of the C++ programming language to implement solutions	Problem-solving Quizzes Case Studies Presentations Group Discussions Projects	Three lecture Periods (9 hours)	• Teacher writes code in the classroom while the students follow. • Students writes code (outside classroom). • Teacher to guide students while students write code in the classroom. • Teacher guides students while students solve problems in groups. • Teacher to use Technology Enhanced Learning (TEL) Tools (e.g. Padlet, Socrative). • Teacher to flip teaching (i.e. teacher to share video resources to students to watch via the LMS and report while the teacher leads the discussions).

The suggested reading materials recommended for the course include:

1. Malik, D.S. (2013) C++ Programming: From Problem Analysis to Program Design (6th Edition)
2. Stanley B. Lippman, Josée Lajoie, and Barbara E. Moo (2018) C++ Primer (5th Edition)
3. Bjane Stroustrup (2016) Programming: Principles and Practice Using C++ (2nd Eition)
4. Downey, A. (2015) Think Python. Second edition. Beijing: O'Reilly

In summary, the researcher developed the curriculum mediation for activity-based computer programming instruction within a blended learning environment for the three HEIs in Ghana. The curriculum was implemented and used by the HEIs in semesters 2 and 3 respectively. In the next section, a web-based computer programming instructional platform was likewise developed and implemented to validate the relevance of the developed curriculum amidst COVID-19 in teaching computer programming.

5.4 Development of the Activity-Based Programming Instructional Platform (ABPI)

The activity-based learning platform is a web-based learning management system designed for teaching and learning in HEIs with the prime focus of using an activity-based instructional approach to teach computer programming courses. It has both students' and instructors' interfaces. In another vein, the researcher also developed an algorithm and a GUI interface (Nkwo, Orji, and Ugah 2021) that serves as a mediating tool to enhance activity-based teaching and learning. This is different from the LMS platform used by the universities to facilitate the blended learning approach.

The development of the ABPI system emanated from COVID-19 measures to equally implement an activity-based approach virtually where distance was not a barrier for teaching and learning of computer programming and thus, to validate the relevance of the activity-based learning approach in teaching computer programming either face-to-face or remotely for institutions with or without an LMS.

The researcher saw the need for developing the ABPI platform to enhance the teaching and learning of computer programming using an activity-based approach as it provides a hands-on approach of learning computer programming. The developed system was tested and used by 124 users with 4 instructors and 4 courses among the three HEIs.

The system was developed based on a requirement engineering and critical system analysis. The requirement analysis phase was conducted among the Instructors, three Heads of Department and 12 selected students from levels 100 to 400. After the requirement analysis of the system, the researcher developed the flow chart as indicated in Figure 5.2. The following chart is the diagrammatic representation of the algorithm to solve the problem of the study.

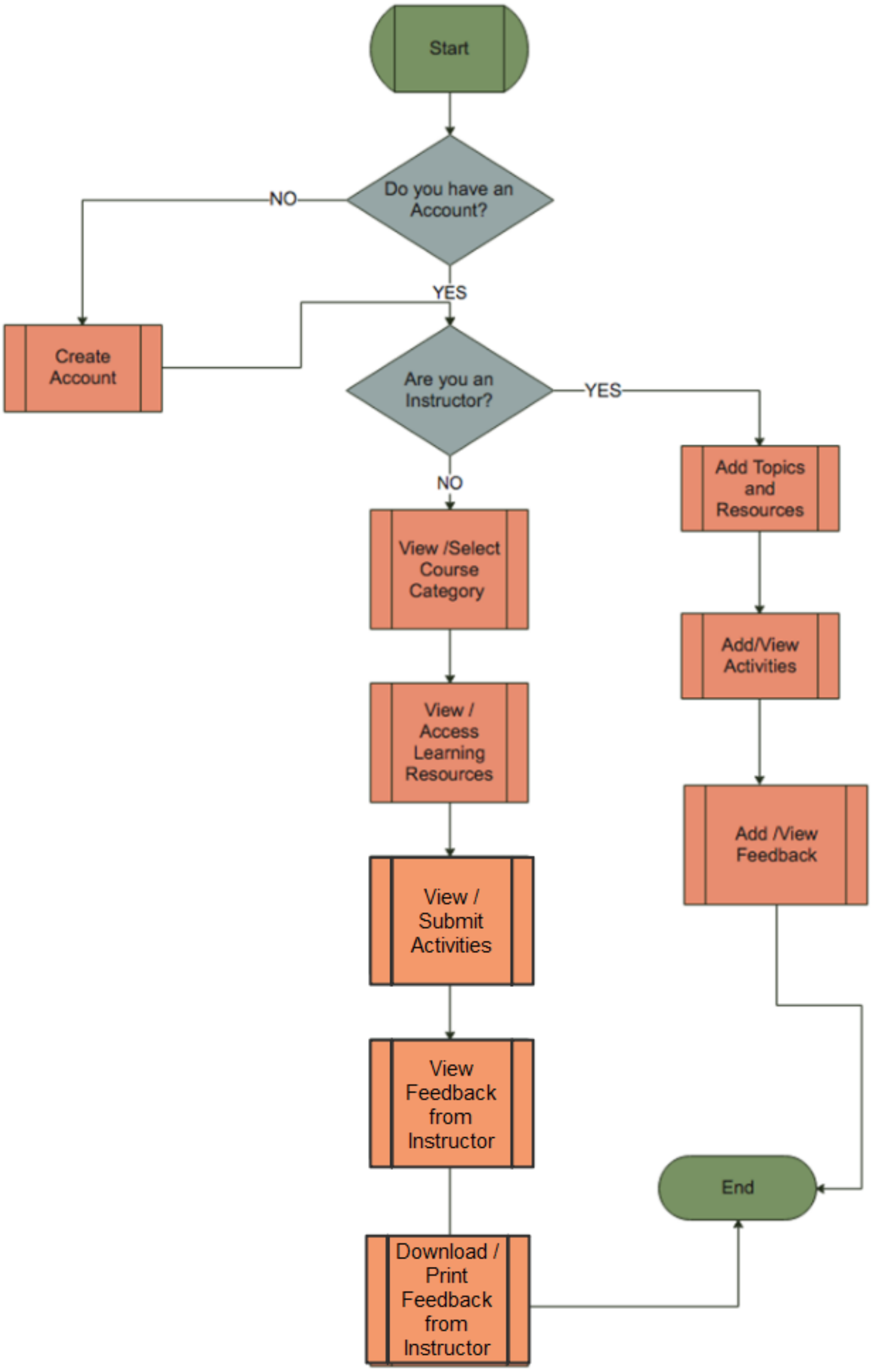


Figure 5.2: Flow Chart of the ABPI System

The flow chart of the system starts by asking users to either create account or sign in to the system. By default, new users of the system are requested to register before access will be granted to them. Upon a successful login, the system decides to grant access to users who are mainly students and lecturers to the dashboards of the systems. At this juncture, students see students' dashboard (e.g. view course, access learning resources, submit assessments, and view feedback from lecturers). On the other hand, instructors see the lecturers' dashboard with the following examples: add topics, resources, activities and view or add feedbacks. Upon completing all activities, the user (student or instructor) is required by the system to close or end.

The use case diagram of the system depicting the system boundaries and usability indicates three major components (Figure 5.3). These include the students', lecturers' and administrators' boundaries. The students' boundary grants them access to view course resources and activities, submit assessments, and view feedback from lecturers. The instructor's boundary grants access to the lecturer to add or view topics, resources, activities and grant feedbacks to students. The administrator's boundary of the system provides a total access to the system. Thus, the administrator of the system can add, view, update or delete a user, courses, topic, resources, activities and feedback.

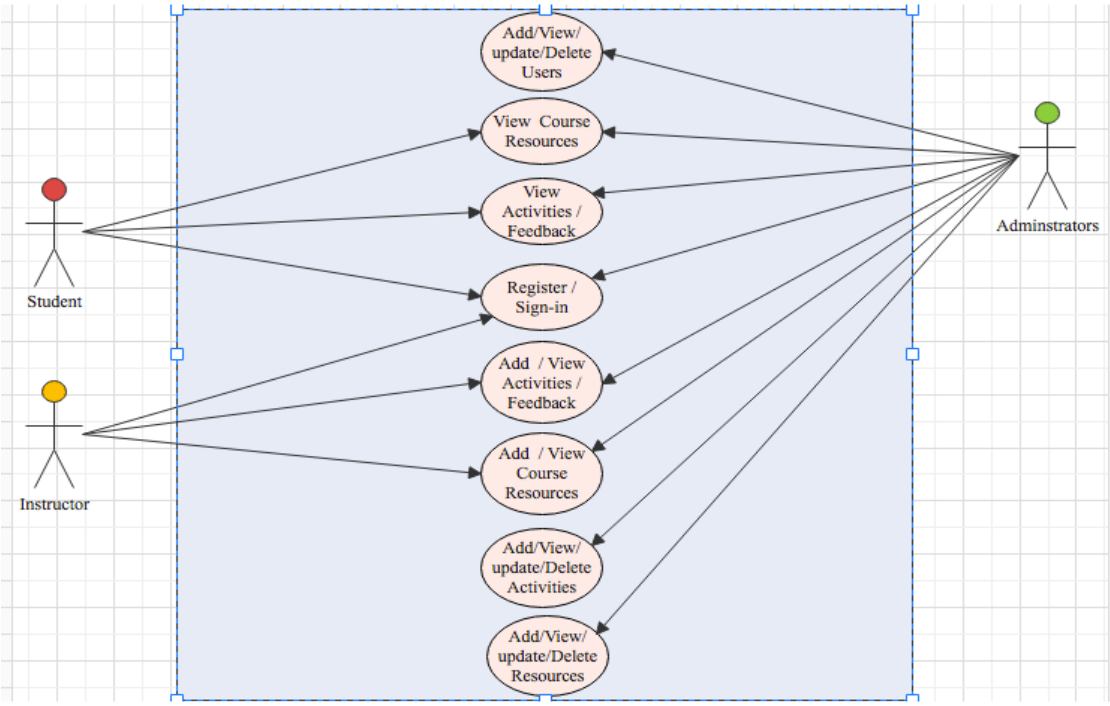


Figure 5.3: Use Case Diagram of the ABPI System

5.4.1 User Interface Design of ABPI System

According to Nkwo, Orji and Ugah (2021) in recent years, Human-Computer Interaction (HCI) research has gained popularity in Africa. Researchers and designers have taken use of the opportunity afforded by developments in mobile and web technologies and increased community access to internet services to develop and implement user-centered digital solutions aimed at African audiences and tackling African challenges. The assertion of Nkwo, Oriji and Ugah affirms the motivation of the researcher to develop a user-centered application that enhances students and instructors teaching and learning interactivities via a web-based system.

The human interface system was developed using the Rapid Application Development Approach (RAD). The user interface design used the Model View Controller (MVC) approach with PhP programming language and PhPmysql as the database backend. The platform runs on all web browsers and smart phone devices. The user interface of the system has both the front-end (i.e., Students' and Instructors' interfaces) and a back-end (Administrator's interface). The system has the following user interface functionalities;

1. A login and user registration interface to authenticate or register users to a class
2. Add Activities interface for the lecturers
3. Start Activity interface for the students (students are required to submit the activities upon completion which is time-bound).
4. A multimedia-enabled interface to record an activity and share (i.e. audio recording, video recordings and documents). The functionality is for both the lecturer and the student.
5. A Resources interface, where instructors can upload (PPTs, PDFs, and video tutorials, YouTube link, etc.) on each topic.
6. A feedback interface to grant feedback on students' activities on assessment.
7. An administrative dashboard that generates reports on all activities in the system.

The dashboard of the Administrator interface is shown in Figure 5.4 below.

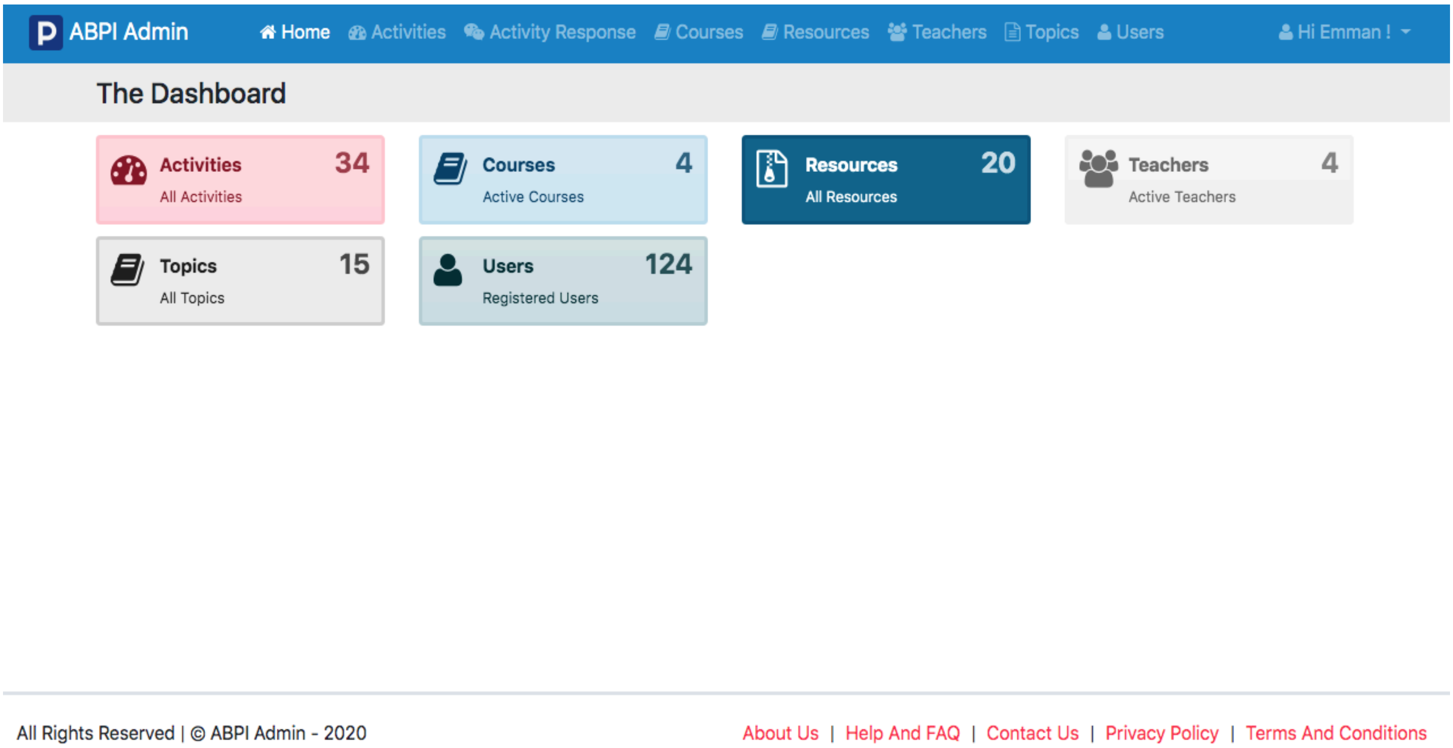


Figure 5.4: Dashboard of the ABPI System (Administrator Interface)

The administration interface which serves as the back-end of the system keeps data and summaries of all the activities that happen on the platform. The system has six major activities on the platform:

1. The Activities interface
2. Courses interface
3. The Topics
4. The Resources interface
5. Records on the Teachers
6. Records on Users

5.4.2 The Activities Interface

The Activity interface provides privilege to an instructor or an administrator to add, edit or delete an activity. The prime motive of this interface for teaching and learning is to provide the platform for posting activities like assignments, quizzes, and set-up a threaded forum discussion. Upon access to the system, students can then submit responses to the activities. A sample activity interface is shown in Figure 5.5.

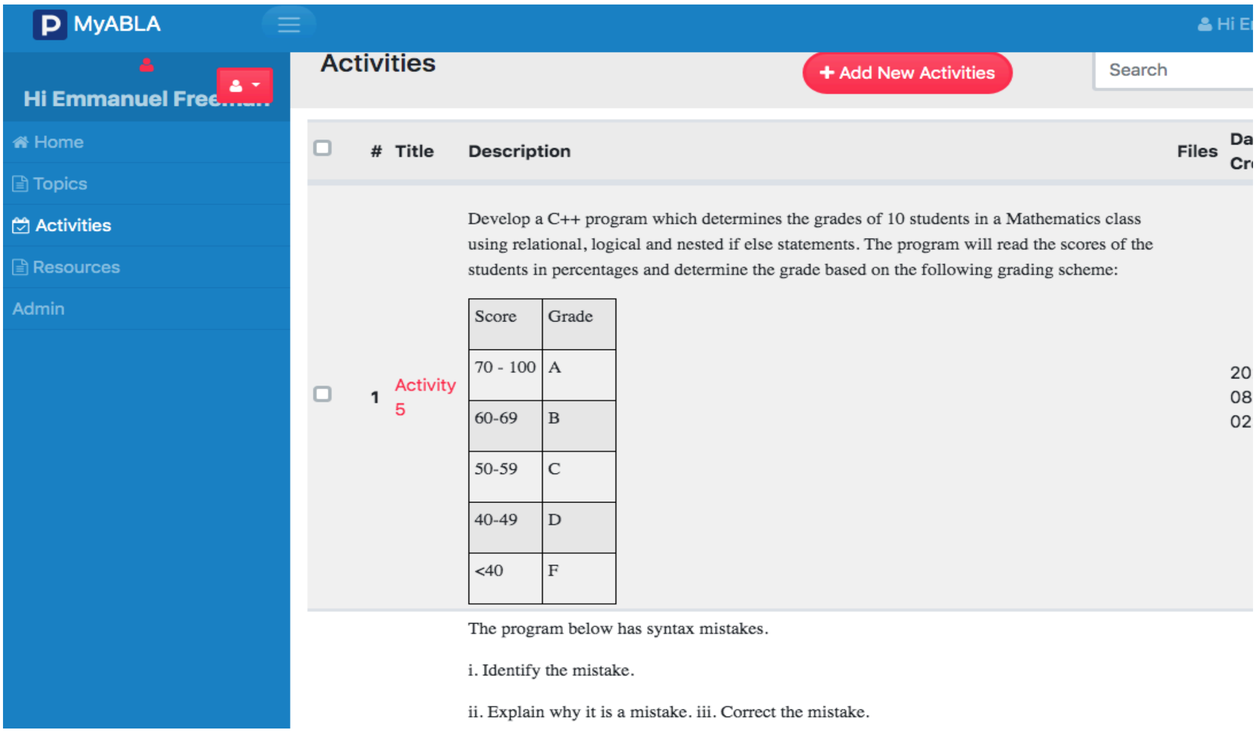


Figure 5.5: Sample Activities

When the instructor accesses the activity interface, he or she can further view all the submissions made in response to the activity posted on the system which is time bound. All late submissions are flagged as late submissions, especially in the case of quizzes and assignments submissions.

An analysis of the activities performed on the system after launching the ABPI system among the 3 HEIs indicated 34 different activities as demonstrated on the dashboard interface in Figure 5.4 above. Further probing of the system revealed that, among the 4 instructors enrolled on the system, at least each lecturer posted 7 or more activities within 6 weeks after the launch of the system.

5.4.3 The Courses and Topics Interface

The courses and topics interface also grants the privilege to add, edit and delete a topic to a course. Once a course is set-up on the "ABPI" System, instructors upon registration, can select the course(s) they are supposed to teach and consequently, add the topics under each course. A sample interface is demonstrated in Figure 5.6.

An examination of the ABPI system showed that 15 topics and four courses were posted after the system was rolled out. The four courses running on the system included: Introduction to Programming (HEI-1), Programming with C++ (HEI-2), Java Programming (HEI-3) and Microprocessor System Interfaces (HEI-1). Major topics covered in the C++ Programming for instructors in HEI-1, HEI-3 and HEI-2 on the system included; Introduction to Programming, Algorithms, Flowchart and Pseudocodes, Programming Theory: Boolean Logic, Arrays, Sorting and Searching, Pointers, and Functions.

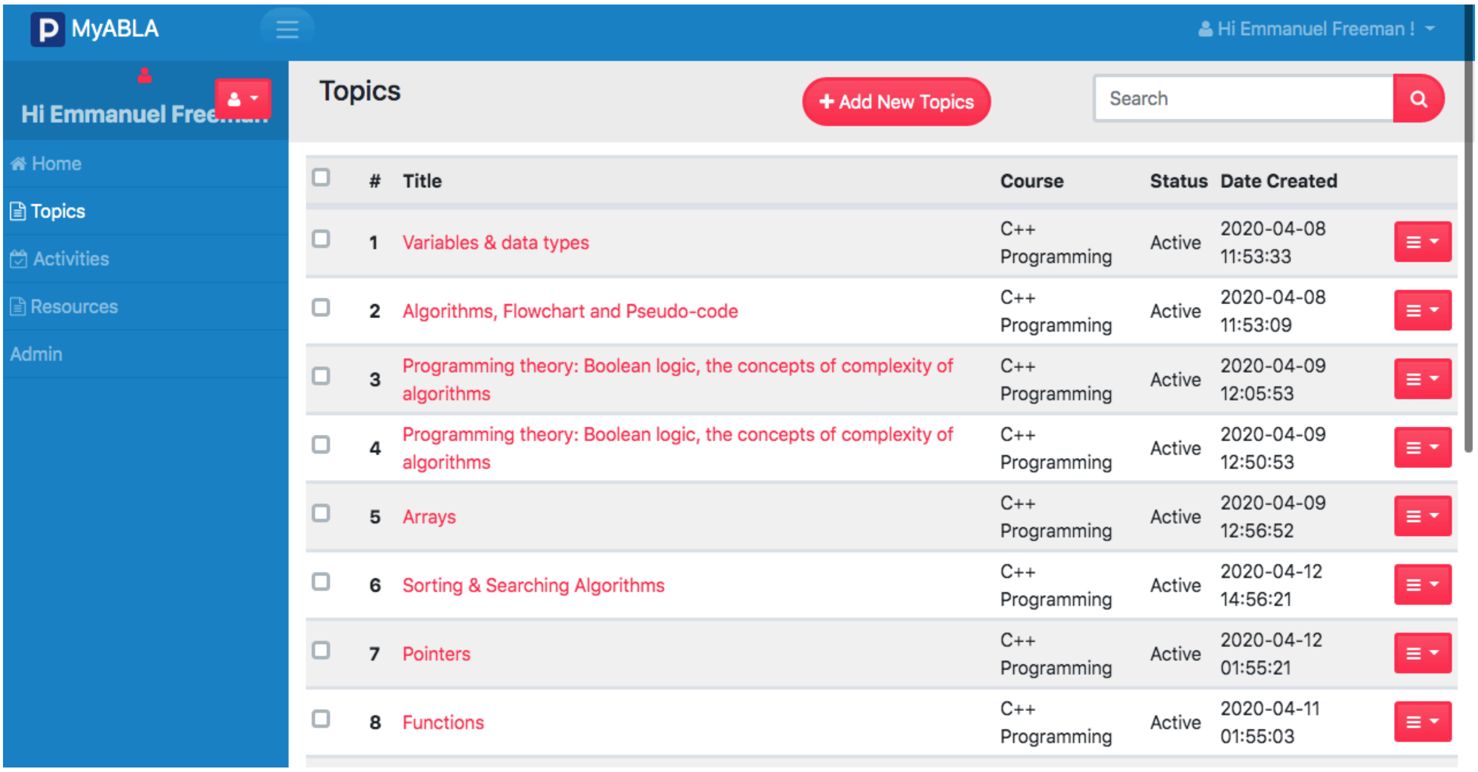


Figure 5.6: Sample Topics and Courses

5.4.4 Enrolled Instructors and Courses

The system upon implementation had four instructors from the three HEIs enrolled on it. The system was initially accessible to three instructors within the three HEIs who are noted to be subject matter experts in teaching programming who reviewed and tested the system before implementation. Interestingly, another instructor in HEI-1, who also reviewed and tested the system made the system available to his engineering course (Microprocessors Systems and Interface) and employed an activity-based instructional approach. It was noted from the instructors who used the system that ABPI platform was very useful to their teaching and learning experiences. The instructors enrolled on the system are indicated in Figure 5.7.

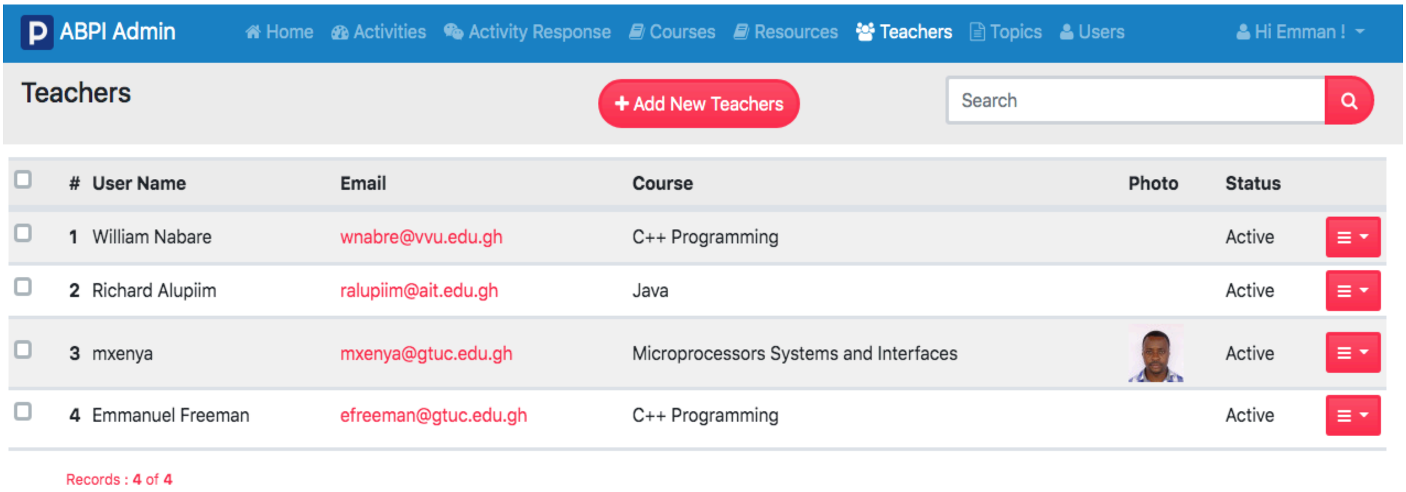
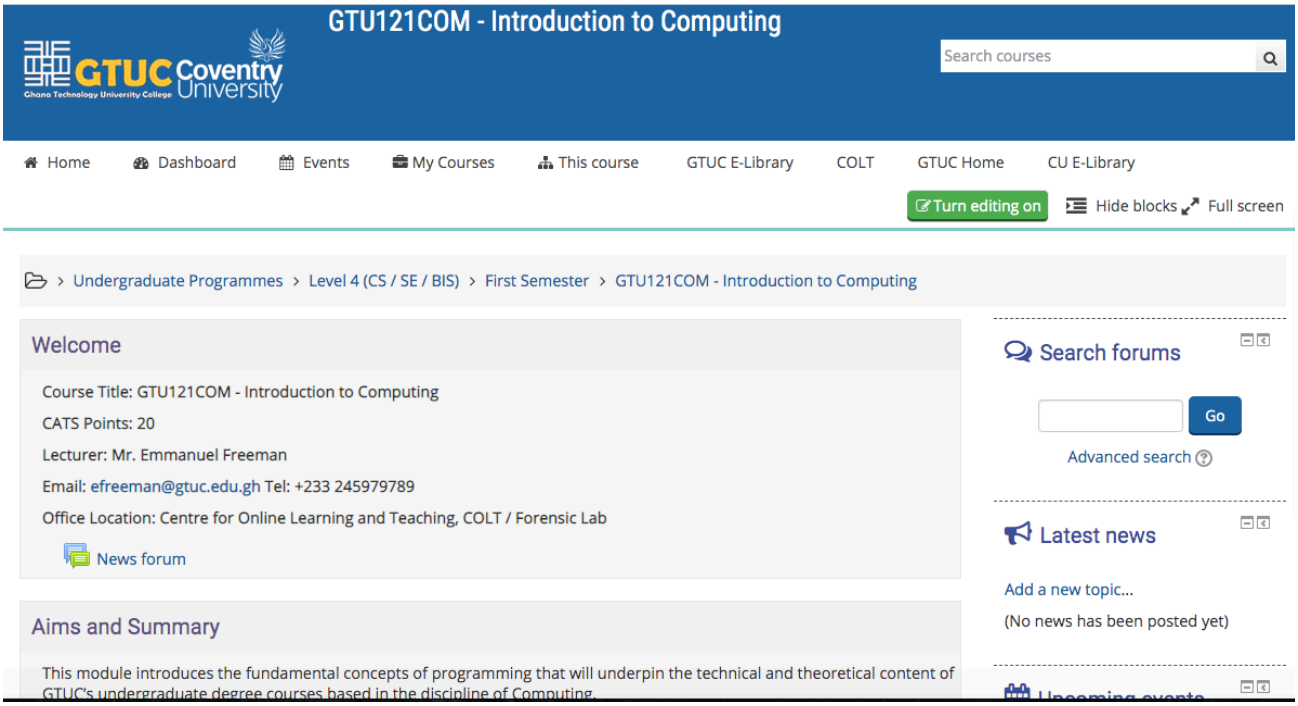


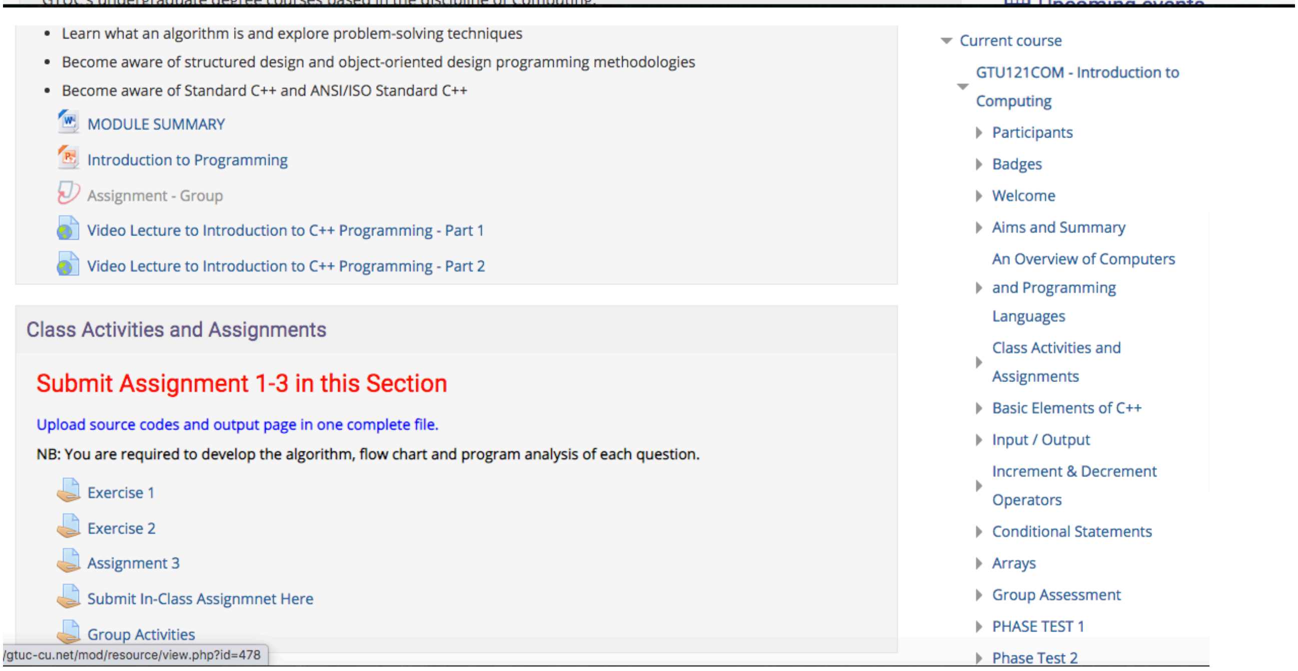
Figure 5.7: Enrolled Instructors and Assigned Courses

5.4.5 Learning Management System (LMS) Used

For the purpose of this research, Moodle 3.4 as the LMS was used, the blended learning approach mostly used for teaching and learning was adopted (Coates, James and Baldwin, 2005; Dalsgaard, 2006). As the Head for the Centre for Online Learning and Teaching (COLT) at the researcher's institution, he ensures that all teaching and learning employ both face-to-face and online delivery, i.e. blended learning approach. This is the cardinal point for using an LMS for the asynchronous delivery of the computer programming course used among all the three HEIs (Matarirano et al., 2021).

The LMS platform provided activities (e.g. assignments, forum discussions, quizzes, Turn-it-in submissions, group chatting etc.) and resources that promote teaching and learning of students. Learning resources accessible to students included text, files (e.g. PDF, DOCX, DOCS, PPT), multimedia resources (e.g. videos, audio, podcast, etc.), links to Universal Resource Locators (URLs), etc. The LMS used for the teaching and learning of computer programming in HEI-1, is indicated in Figure 5.8.


GTU121COM - Introduction to Computing
GTUC Coventry University
Ghana Technology University College
Search courses
Home
Dashboard
Events
My Courses
This course
GTUC E-Library
COLT
GTUC Home
CU E-Library
Turn editing on
Hide blocks
Full screen
Undergraduate Programmes > Level 4 (CS / SE / BIS) > First Semester > GTU121COM - Introduction to Computing
Welcome
Course Title: GTU121COM - Introduction to Computing
CATS Points: 20
Lecturer: Mr. Emmanuel Freeman
Email: efreeman@gtuc.edu.gh Tel: +233 245979789
Office Location: Centre for Online Learning and Teaching, COLT / Forensic Lab
News forum
Aims and Summary
This module introduces the fundamental concepts of programming that will underpin the technical and theoretical content of GTUC's undergraduate degree courses based in the discipline of Computing.
Search forums
Go
Advanced search
Latest news
Add a new topic...
(No news has been posted yet)


Learn what an algorithm is and explore problem-solving techniques
Become aware of structured design and object-oriented design programming methodologies
Become aware of Standard C++ and ANSI/ISO Standard C++
MODULE SUMMARY
Introduction to Programming
Assignment - Group
Video Lecture to Introduction to C++ Programming - Part 1
Video Lecture to Introduction to C++ Programming - Part 2
Class Activities and Assignments
Submit Assignment 1-3 in this Section
Upload source codes and output page in one complete file.
NB: You are required to develop the algorithm, flow chart and program analysis of each question.
Exercise 1
Exercise 2
Assignment 3
Submit In-Class Assignmnet Here
Group Activities
/gtuc-cu.net/mod/resource/view.php?id=478
Current course
GTU121COM - Introduction to Computing
Participants
Badges
Welcome
Aims and Summary
An Overview of Computers and Programming Languages
Class Activities and Assignments
Basic Elements of C++
Input / Output
Increment & Decrement Operators
Conditional Statements
Arrays
Group Assessment
PHASE TEST 1
Phase Test 2

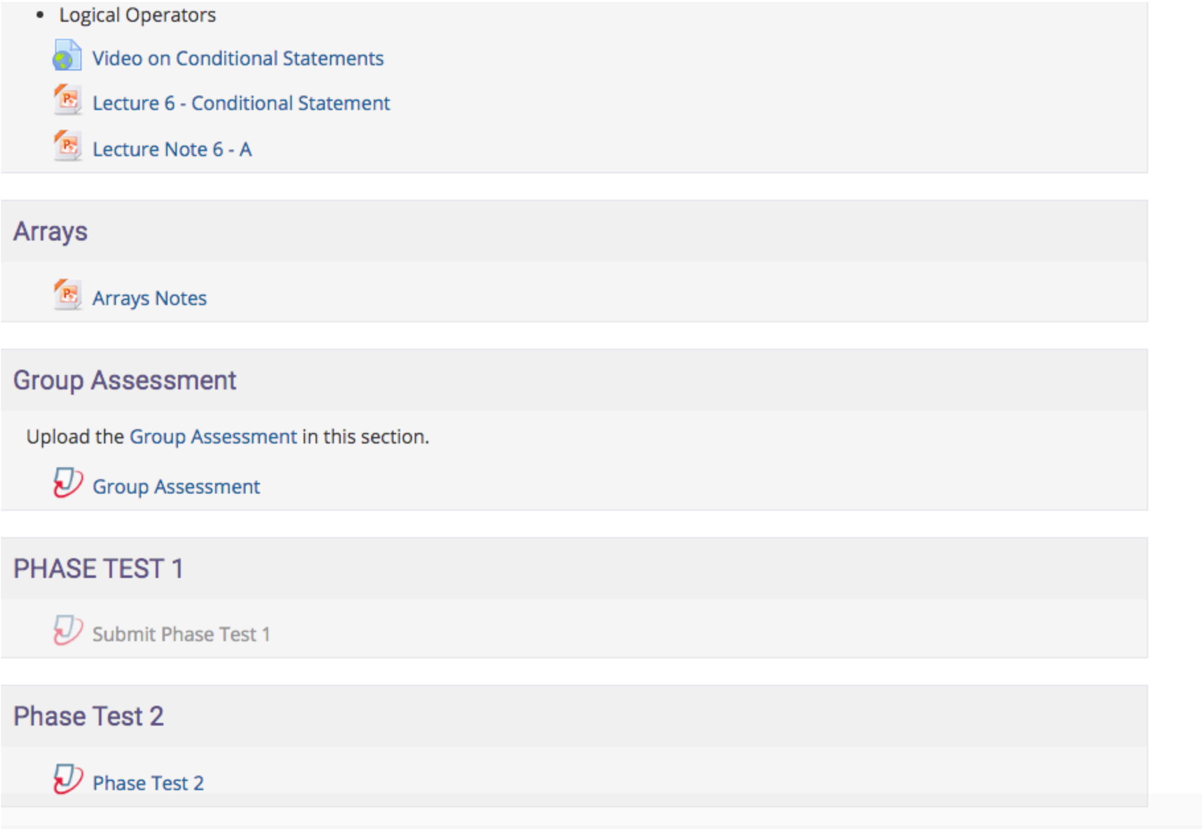


Figure 5.8: Sample LMS Platform Used

5.5 Analysis of the Curriculum Mediations Developed

The analysis of the curriculum mediations that support teaching and learning of programming within a blended learning environment was conducted and six instructors among the three HEIs were interviewed. Generally, the action research approach employed for this study sought to involve the researcher in the study, hence the researcher is a testament to the effectiveness of the curriculum developed for the instruction of computer programming within a blended learning environment using an activity-based approach. The results from the interview conducted per the interview guide in Appendix D refers:

1. Suitability of the Curriculum

It was noted from the interview responses from the instructors that the curriculum developed was suitable for Computer Science foundational courses with five instructors responding in affirmative.

Instructor A – HEI-1: "*The curriculum is perfect for first year computing degree programmes*".

Instructor C – HEI-3: "*The curriculum is good for teaching computer programming*".

Instructor E – HEI-2: "*The activity-based application component for the design of the curriculum is just awesome. Students needs to be engaged with hands-on activities to develop systems*"

2. The Learning Outcomes

On the learning outcomes, all the six instructors interviewed admitted that they were lucid and relevant to the teaching and learning of programming.

3. Pedagogic Approach

The use of the activity-based approach together with the ABPI develop was highly recommended by five instructors and confirmed that it should be encouraged for all computing science courses. This was established during the interview sessions with the instructors.

4. Assessment Practices

The assessment methods for teaching and learning of programming was also keenly considered. During the interview sessions with the six programming instructors, four of them responded in affirmative that a formative assessment method (Price and Kirkwood, 2011) should carry greater assessment percentage (e.g. 70% or 60%) of the total course assessment while the remaining 30% or 40% carries the summative assessment. A further probing to ascertain the instructor's reasons for choosing formative assessment over summative assessment was that the formative assessment was seen to be useful to prepare students using a series of activities, assignments, group work, individual projects, etc. to equip the students with hands-on experience of coding.

Also, the formative assessment was seen to prepare the students for summative assessment and give students an early indication of their progress towards the intended learning outcomes of the course. Four out of six of the interview lecturers responded in affirmative that the formative assessment practice is a good approach to prepare the students to pass all forms of summative assessment in programming classes. On another hand, two instructors preferred the use of the summative assessment (e.g. final examination, class test, quizzes, etc.) to cover a greater assessment percentage over the formative assessment (Chew, Jones and Turner, 2008; Olsson and Mozelius, 2016).

5.6 Reflections on the Curriculum Mediation and ABPI for Computer Programming Instruction.

Following the action research used for the study, the researcher at the 5th and 6th stage (i.e. second and third semesters), conducted a personal reflection on the findings received from both the study and the developed curriculum and the ABPI. The researcher sought advice from his Head of Department (HOD) of the Computer Science of the researcher's institution and the six instructors in the three HEIs to test the designed curriculum and the ABPI system.

Firstly, the researcher took interest to measure how the teaching and assessment practices affected the student's performance. Initially, the researcher proposed the use of greater percentage of formative assessment (i.e. 60% or 70%) and 40% or 30% as the summative assessment for the courses. As a member of the Academic Board and Faculty Board of the researcher's institution, the 60% formative and 40% summative was considered following a significant improvement on the students' performance in computer programming during the two semesters in which the curriculum was being used compared to the previous years' performances. The improvement of students' performance in programming was also evident among the other lecturers in all the three HEIs teaching programming after implementing the proposed curriculum or course syllabus.

The researcher also noticed from both the qualitative and quantitative data that the activity-based approach affected the programming skills of the students positively. From a personal observation as an instructor and feedback from colleagues in the three HEIs, solidly affirmed that the activity-based approach for teaching and learning programming within the blended learning environment supported the students' cognitive and programming skills. Correspondingly, most of the students equally saw the activity-based approach as an effective approach to teaching and learning programming.

Again, the researcher saw the activity-based pedagogic approach for teaching other computing science courses as a remedy to enhance the students' capabilities of solving real-world problems and consequently increase their employability opportunities after they earn their degrees.

5.7 Relevance of the Second Semester Findings on the ABLA Framework Development

During the second semester of the semester AR, the researcher developed a curriculum that mediates an effective teaching and learning of computer programming within a blended learning environment. The activities in the second semester aided the researcher to implement an activity-based computer programming curriculum for HEIs in Ghana. The findings also aided the researcher to develop an ABPI platform which validated the use of activity-based instructional approach within a BL environment. The findings contributed about 50% of the action research towards the development of the ABLA framework. This is represented in Figure 5.9 below. The remaining third semester of the AR is discussed in the subsequent chapters.

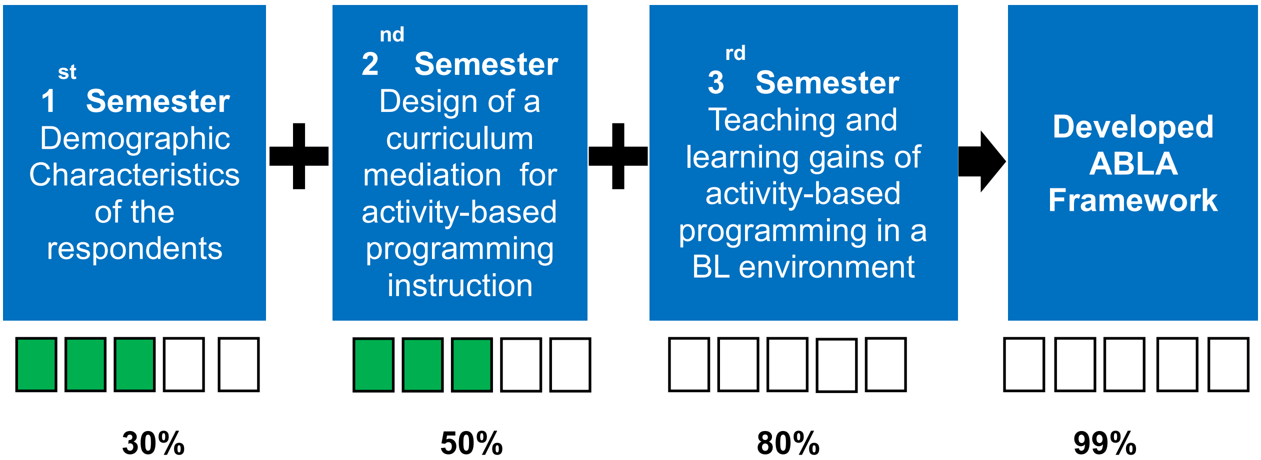


Figure 5.9 Progress View Towards ABLA Framework Development as at Semester 2

5.8 Conclusion

Prior to the development of the curriculum for the teaching and learning of computer programming, the demographic details of the respondents, the activity-based approach used and the challenges faced in teaching and learning of computer programming in HEIs Ghana were analysed in this chapter. In pursuance to the development of the activity-based framework, the researcher consequently developed a curriculum that mediates an effective teaching and learning of computer programming within a blended learning environment. The implementation of the curriculum mediation was used in semesters 2 and 3 and was finally accepted among instructors who teach programming in the three HEIs upon several reflections by the researcher and the instructors in the three HEIs in Ghana. The revised curriculum guided the teaching and learning of computer programming and consequently, the researcher designed an activity-based programming instruction application to enhance teaching and learning. The curriculum and the ABPI system were seen to be very relevant in teaching and learning of programming in the three HEIs.

Findings from the interview discussions with six computer programming instructors on their experiences of the curriculum and the activity-based instructional system for computer programming were ascertained and the conclusions proved very positive and relevant. The researcher finally performed a reflection on the study.

CHAPTER 6

DEVELOPMENT OF THE FRAMEWORK FOR ACTIVITY-BASED COMPUTER PROGRAMMING INSTRUCTION

6.1 Introduction

In the previous chapter, the researcher developed a curriculum mediation for activity-based computer programming instruction within a blended learning environment in HEIs in Ghana. In pursuance of the main aim of this study to develop a framework for activity-based programming instruction within a blended learning environment of Higher Education Institutions (HEIs) in Ghana, an initial conceptual design was developed and was implemented based on the developed curriculum used for computer programming instruction in semesters two and three of the action research. Data captured in this study has been analysed and the results obtained from the data collected are presented in this chapter.

Data was collected from the students and lecturers per the action research approach. The SPSS 26 software was used for the preliminary analysis, model evaluation and final analyses. Additionally, AMOS 26 and Atlas Ti software were used for the quantitative and qualitative data analysis respectively. Quantitative and qualitative analysis were performed towards the design of the ABLA framework for teaching computer programming in HEIs.

6.2 Results of Third Semester Activities of the Action Research

After establishing the demographics, the current activity-based approach used in teaching computer programming, the prevailing challenges faced by students in the comprehension of computer programming in the first semester, the researcher designed and implemented the curriculum for computer programming instruction during the second semester.

Further, at the end of the second semester the researcher collected new data based on the conceptual framework to ascertain how the constructs (i.e. cognitive activities, social activity, instructional activities, activity-based approach used, skills development, learning gains, and assessment and feedback) relate to each other in the developed framework. See Appendix E for the instrument used to collect all the quantitative data after semester two and Appendixes

F and G for the instruments used to collect the qualitative data of the AR in the third semester towards the development of the ABLA framework. The characteristics of the constructs and the analysis of the data collected after the second semester can be seen in the subsequent sections.

6.2.1 Characteristics of the Contructs

In order to perform further testing of the statistical data, the researcher took interest to analyse the characteristics of the constructs used for the study. In total, there were seven (7) constructs in the conceptual framework of the study. This consisted of cognitive activities, the activity-based approach used, social activities, instructional activities, skills development, learning gains and assessment and feedback. See Appendix E for the instruments used to collect the data. The characteristics of the constructs is indicated in Table 6.1.

Table 6.1 Characteristics of the Constructs

Constructs	**Mean**	**Median**	**95% Confidence Interval for Mean**		**Std. Deviation**	**Skewness**	**Kurtosis**
			Lower Bound	Upper Bound			
Cognitive Activities	3.5842	3.7500	3.4822	3.6861	.89720	-.559	.076
Activity-based Approach Used	2.3167	2.0000	2.1968	2.4365	1.05484	.598	-.551
Social Activities	3.6089	3.6667	3.5192	3.6986	.78942	-.881	1.081
Instructional Activities	3.4867	3.6250	3.3724	3.6009	1.00554	-.530	-.265
Skills Development	3.4433	3.6667	3.3412	3.5455	.89889	-.594	.142
Learning Gains	3.4222	3.5000	3.3215	3.5229	.88616	-.551	-.005
Assessment and Feedback	3.5811	3.6667	3.4824	3.6798	.86881	-.554	.248

It was found that the mean and median responses of all the constructs were more than 3.4 and 3.5 respectively, except for the activity-based approach used which gave a mean response of 2.3167 and median of 2.0000 among the three HEIs. This implies that the students mostly agreed and responded in affirmative that the six constructs frequently apply to them except the activity-based approach based on the Likert scale. The reason for the low mean response of the activity-based approach used denotes the fact that there were different lecturers with different teaching styles, different instructional environments, and different implementation strategy of the activity-based programming instructional approach as per the curriculum adopted. Again, the confidence interval for the means using 95% confidence indicates the lower bound and upper bound of all the constructs were above 3.3 and 3.5 except the activity-based approach used which gave 2.1968 (lower bound) and 2.4365 (upper bound). Also, the skewness of the data proved a negative skewness of all the constructs except the activity-based approach used which indicated a positive skewness of .598 and Kurtosis value of -.551 as depicted in Table 5.6.

1.2.2 One-Way ANOVA Test to Determine Differences in Mean Scores by the HEIs Based on the Constructs

A one-way ANOVA test was conducted to test if there were any statistical differences among the three institutes in terms of the seven major constructs derived from the conceptual framework, i.e., the activity-based approach used, cognitive activities, social activities, instructional activities, activity-based instructional approaches in teaching programming, skills development, feedback and learning gains as indicated in Table 6.2. It was noted that there existed statistically significant differences of $p=0.00$ among the three institutions as indicated in Table 6.2. Thus, since $p < 0.05$, then it is established that there is a statistically significant difference between the three HEIs. This is to affirm that the findings from the three HEIs were different from each other with respect to the activity-based approach used, cognitive activities, social activities, instructional activities, skills development, feedback and learning gains among the students in programming classes.

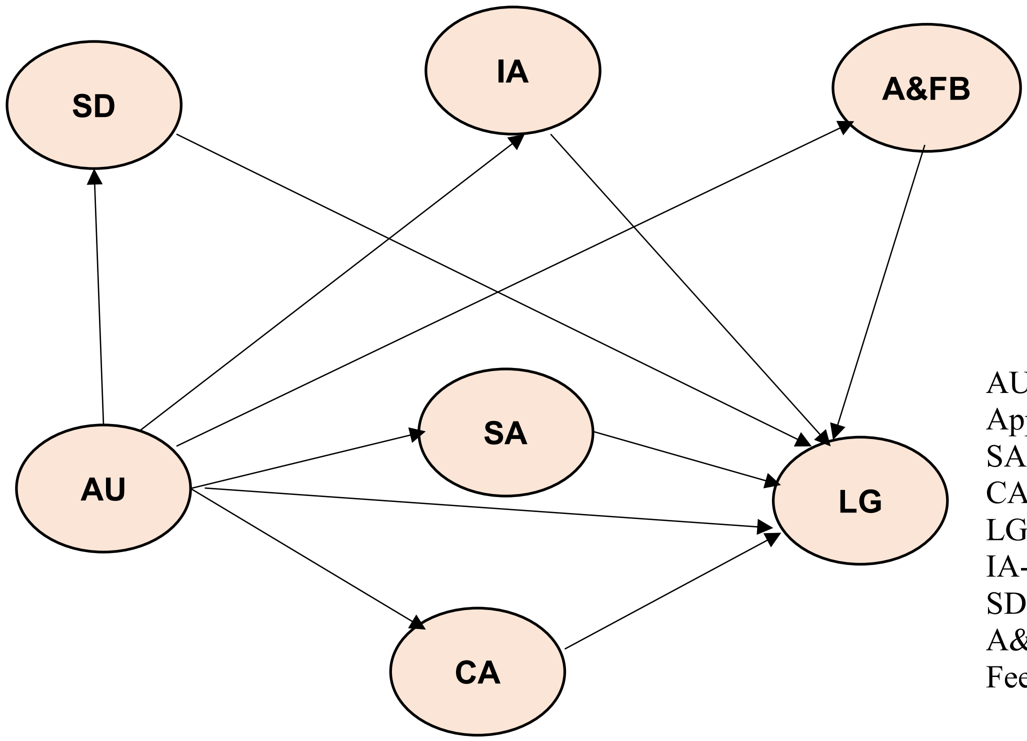
Table 6.2 One-Way ANOVA Test to Determine Differences in Mean Score by the HEIs

Constructs	Sum of Squares	Df	Mean Square	F	Sig.
Cognitive Activities	40.375	2	20.188	29.932	.000
	200.312	297	.674		
	240.687	299			
Activity-based Approach Used	48.427	2	24.214	25.298	.000
	284.267	297	.957		
	332.694	299			
Social Activities	28.827	2	14.413	27.179	.000
	157.505	297	.530		
	186.332	299			
Instructional Activities	69.229	2	34.614	44.105	.000
	233.093	297	.785		
	302.322	299			
Skills Development	28.794	2	14.397	20.094	.000
	212.798	297	.716		
	241.592	299			
Learning Gains	29.738	2	14.869	21.536	.000
	205.059	297	.690		
	234.796	299			
Feedback	50.082	2	25.041	42.350	.000
	175.611	297	.591		
	225.693	299			

6.3 Designing the ABLA Framework

The researcher took interest to ascertain all the constructs that enhance quality teaching and learning of computer programming using an activity-based instructional approach within a blended learning environment. The various compositions of the designed framework include instructional activities, cognitive activities, social activities, skills development, assessment and feedback, the activity-based approach used and learning gains. The conceptual design of the framework is seen in Figure 6.1 below. The researcher followed the following steps during the AR to develop the framework.

1. Develop the conceptual framework based on the existing literature and experts' opinion on the teaching and learning of computer programming within a blended learning environment. The conceptual framework was validated by experts in the areas of CS, IS and IT and have got interest in programming instruction.
2. Collect data on the respondents' (Students and Instructors) demographic information, their experiences on the current activity-based approach used in teaching programming, and the challenges in the teaching and learning of programming among the three HEIs in Ghana during the first semester of the AR.
3. Develop a curriculum mediation that could support the teaching and learning of computer programming.
4. Implement the curriculum developed in semester two based on the seven constructs.
5. Develop and implement an ABPI platform to validate all the constructs that supports the teaching and learning of computer programming (e.g. instructional activities, social activities, cognitive activities, assessment and feedback, the activity-based approach used, and the learning gains or satisfaction).
6. Collect Data in Semester 2 on the implemented curriculum and ABPI platform to ascertain the relevance of the curriculum in programming instruction per the constructs.
7. Collect a new data in semester three to test each of the constructs of the framework using structural equation modelling to ascertain the relevance of the constructs on the framework in semester three.



Abbreviations

AU- Activity-based Approach Used
SA- Social Activities
CA- Cognitive Activities
LG – Learning Gains
IA- Instructional Activities
SD – Skills Development
A&FB – Assessment and Feedback Activities

Figure 6.1 Conceptual Design of the ABLA Framework

Based on the conceptual design of the ABLA framework, the following research hypotheses were outlined to guide the study:

Research Hypotheses

1. H_1: The activity-based teaching and learning approach of computer programming in HEIs in Ghana within a blended learning environment include but not limited to case studies, quizzes, projects, group discussions and presentations, problem-based learning and concept mapping.
2. H_2: The new developed curriculum used in teaching computer programming within a blended learning environment supports activity-based learning.
3. H_3: The activity-based learning approach of teaching programming within HEIs in Ghana has a significant positive effect on students' learning gains.
4. $H_{3.1}$: The activity-based learning approach of teaching programming within HEIs in Ghana has a significant positive effect on students' cognitive development.
5. $H_{3.2}$: The activity-based learning approach of teaching programming within HEIs in Ghana has a significant positive effect on students' social activities and learning.

6. $H_{3.3}$: The activity-based learning approach of teaching programming within HEIs in Ghana has a significant positive effect on students' teaching activities.
7. $H_{3.4}$: The activity-based learning approach of teaching programming within HEIs in Ghana has a significant positive effect on students' engagement.
8. $H_{3.5}$: The activity-based learning approach of teaching programming within HEIs in Ghana has a significant positive effect on students' feedback and assessments.
9. $H_{3.6}$: Social activities have a significant positive effect on learning gains among students in HEIs in Ghana.
10. H_4: The developed framework has a significant positive effect on the activity-based learning approach in a blended learning environment for instructing computer programming.

6.3.1 Implementation of the Conceptual Design of theABLA Framework

After the conceptual design of the ABLA framework was developed, the researcher implemented the design based on the developed and implemented curriculum used in teaching computer programming in semesters 2 and 3 respectively. In view of this, the researcher planned, acted, observed and reflected on the design iteratively to develop the framework per the action research.

Colleagues in the three HEIs were consulted and various inputs such as the needs for quality assessment and feedback, learner support, the use of problem-solving activities, TEL tool, etc. were issues considered during the implementation of the developed framework. Consequently, the researcher collected data on the implemented ABLA framework from January 15 – February 20, 2020 before the President of Ghana declared the closure of schools in the second week of March, 2020 as a result of the novel corona virus pandemic (COVID-2019). The instrument that defines all the variables used for collecting the data based on the seven constructs of the ABLA framework for computer programming instruction is also seen in Appendix E.

The mixed method approach was again used to gather data for analysis in the third semester to measure whether or not the proposed designed framework fitted into the teaching and learning of computer programming using the activity-based learning approach within a blended learning environment. Lastly, personal reflections on the developed ABLA framework was performed by the researcher.

6.4 Quantitative Analysis of the Results of the ABLA Framework

Following the action research process, the researcher analysed and evaluated the data derived from the designed and implemented ABLA framework following the mixed method approach of the AR in the third semester. The structural equation model was used to establish the relationships and effects that exist among the constructs (i.e. latent variables) and the observable variables (i.e. the instruments in Appendix E) on the ABLA framework.

A reliability test, preliminary data screening, and validation of the data was performed. An exploratory factor analysis, confirmatory factor analysis, and the analyses of hypotheses using structural equation model (SEM) was conducted on the new data received from the second semester of the action research phases. The SEM was used to test the relationships that exist between the latent and observable variables and to establish the statistical model of the hypotheses.

6.4.1 Reliability of the Instrument

The main issue under reliability analysis was the scale's internal consistency (Salkind, 2018). Thus, it encompasses the degree to which the items that make the scale 'hang together'. One of the most commonly used indicators of internal consistency is the Cronbach's alpha coefficient (α). Ideally, the Cronbach alpha coefficient of a scale should be above 0.7. Bryman and Bell (2015) asserted that Cronbach's alpha coefficient (α) is mainly used to test for all internal consistencies and essentially aid to calculate the mean values of all possible split-half reliability coefficients ranging from 0 (representing not acceptable internal consistency) to 1 (representing acceptable internal consistency).

Jain and Angural (2017) provided the following guidelines where a Cronbach alpha of ≥ 0.9 depicting it is excellent; $0.9 > \alpha \geq 0.8$ is good; $0.8 > \alpha \geq 0.7$ is acceptable; $0.7 > \alpha \geq 0.6$ is questionable; $0.6 > \alpha \geq 0.5$ is poor and < 0.5 is unacceptable. The test was conducted and all constructs were above the required α value of 0.7. The reliability of the scales for this study was satisfied. Table 6.3 below shows the internal consistency of the items in the study.

Table 6.3: Reliability Analysis

Constructs	**No. of Items**	**Cronbach's Alpha**	**Acceptable Level**
Instructional Activities	4	0.934	Excellent
Feedback	3	0.838	Good
Social Activities	3	0.812	Good
Learning Gains	6	0.919	Excellent
Skills Development	3	0.855	Good
Cognitive Activities	4	0.878	Good
Activity Based Learning Used	3	0.824	Good
Total	26	0.866	Good

The outcome of the reliability test conducted attested that all the defined constructs had a Cronbach's alpha coefficient (α) more than 0.7. In fact, instructional activities and learning gains had a reliability value more than 0.9 while all the remaining constructs had the reliability values as $0.9 > \alpha \geq 0.8$. There were 26 items as the instrument with over-all internal consistency value of 0.866 which is acceptable for further tests. According to Jain and Angural (2017), all the reliabilities per the α values are acceptable. Therefore, it affirmed an internal consistency in all the items measured.

5.4.2 Exploratory Factor Analysis

Exploratory Factor Analysis (EFA) is normally done at the early stages of research using SEM as it tries to uncover complex patterns by exploring the dataset and testing predictions (Child, 2006). There are two main factors that determine whether a data set is appropriate for factor analysis. These are sample size and the strength of the relationship among variables (Pallant, 2016).

6.4.3 Validity Test Using Exploratory Factor Analysis

For this study, 90 related items were subjected to exploratory factor analysis using Principal Component Analysis (PCA) as the method of extraction with sample size of 300. This was done using SPSS 26. In determining the appropriateness of the data for factor analysis, Bartlett's test of Sphericity and the Kaiser-Meyer-Olkin (KMO) values must be statistically significant at $p<0.05$ and 0.6, respectively (Pallant, 2016) as indicated in Table 6.4.

Table 6.4: KMO and Bartlett's Test

Kaiser-Meyer-Olkin Measure of Sampling Adequacy		.930
Bartlett's Test of Sphericity	Approx. Chi-Square	23986.360
	Df	3321
	Sig.	**.000**

The results obtained for Kaiser-Meyer-Olkin Measure of Sampling Adequacy was 0.930. Also, the results obtained for the Bartlett's Test of Sphericity obtained an Approximate Chi-Square value of 23986.360, df value of 332 and the p-value of 0.000. Therefore, the research met the standardised criteria and as a result, applying factor analysis was appropriate.

6.4.4 Principal Component Analysis: Communalities

Communalities give information about how much of the variance in each item is explained. The extraction method used for the analysis of the Principal Component Analysis was Oblimin with Kaiser Normalization as the Rotation Method. The Rotation converged in 10 iterations. Low values (less than .3) could indicate that, the item does not fit well with the other items in its components (Pallant, 2016). For this study, it can be seen that, the variables fit well with one another and were suitable for factor analysis based on the fact that almost all the correlation coefficients are above 0.3 as seen in the Pattern Matrix[a] in Table 6.5.

Table 6.5: Pattern Matrix[a]

Factors	Variables	Codes	Component						
			1	2	3	4	5	6	7
Cognitive Activities	Brainstorming and finding relevant information helped me resolve content related questions in the classroom and online	CA10	0.740						
	Combining new information helped me answer questions raised in course activities	CA13	0.700						
	Reflection on course content and discussions helped me understand fundamental concepts in the programming class	CA6	0.591						
	I can apply the knowledge created in this course to my current work or other non-class related activities	CA8	0.457						
Activity-based Approach Used	Educational Games	AU12		0.901					
	Concept Mapping	AU5		0.855					

	Videos	AU6		0.817					
Learning Gains	Overall, I am satisfied with this course	LG1			-0.659				
	I am comfortable with the approach used in teaching programming	LG11			-0.741				
	The online platform is stable	LG12			-0.797				
	The learning environment in the classroom promotes activity- based learning	LG5			-0.777				
	I am satisfied with the instructional approach	LG6			-0.664				
	I am satisfied with the support I gain from my university on my learning experience	LG8			-0.804				
Instructional Activities	The instructor was helpful in guiding the class towards understanding course topics in a way that helped me clarify my thinking.	IA10				-0.824			

	The instructor encouraged course participants to explore new concepts in this course	IA5				-0.850			
	The instructor helped to focus discussion on relevant issues in a way that helped me to learn	IA6				-0.759			
	The instructor helped to keep course participants engaged and participating in productive dialogue	IA8				-0.854			
Social Activities	I am able to form distinct impressions of some course participants	SA2					0.756		
	I felt comfortable conversing through the online medium	SA4					0.872		
	I felt comfortable participating in the course discussions	SA5					0.670		
Assessment and Feedback	Feedback from my instructor is very effective	FB2						0.849	
	The feedback from my peers was very effective	FB3						0.699	

	Assessments, assignments, case studies, projects are well communicated and results or feedback issued on time	FB7						0.748	
Skills Development	I felt like I made a meaningful contribution to the activities in class	SD6							0.782
	I felt challenged by the approach used in teaching the programming course	SD7							0.847
	I felt like I made a meaningful contribution to the activities in class	SD8							0.697

The highest correlation was 0.872. Those below 0.3 were removed from the items for the final analysis. The communality analysis is indicated in Appendix I.

6.4.5 Total Variance Explained

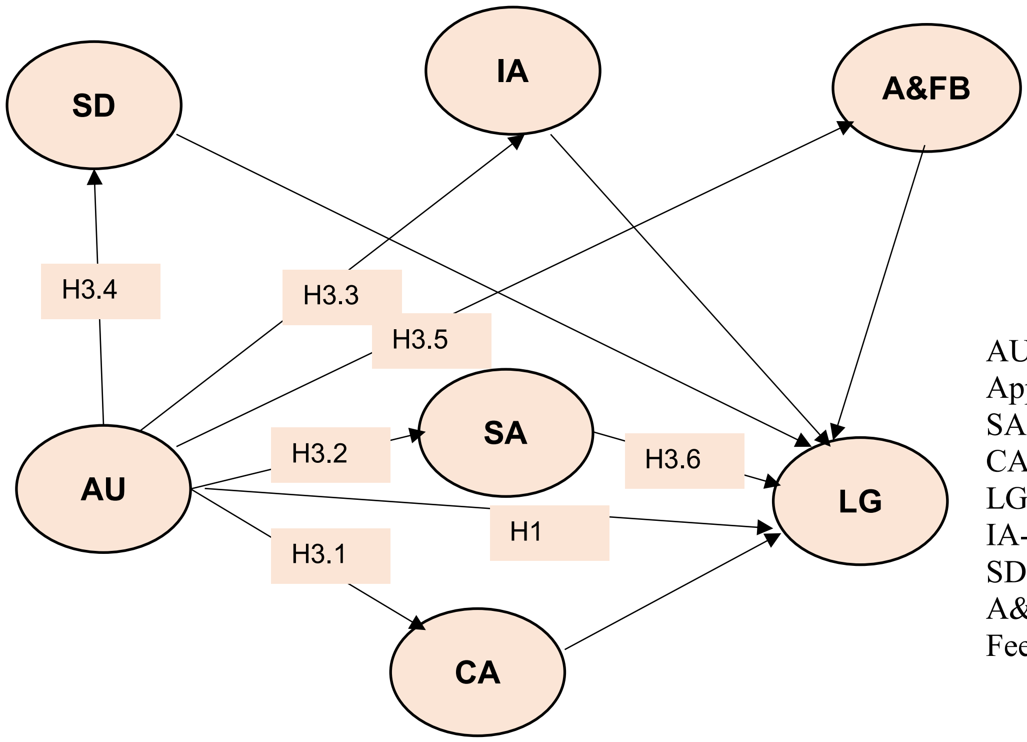
Using Kaiser's criterion, only components that have eigenvalues of 1 or more were of relevance here. In this study, four components had eigenvalues more than 1. However, the extraction was restricted to seven components. These seven components explained a total of 76.71% of the variance (see cumulative % column of the 7^{th} component). When components are correlated, sums of squared loadings cannot be added to obtain a total variance. The Total Variance Explained is indicated in Table 6.6.

Table 6.6: Total Variance Explained

Component	Initial Eigen Values			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings[a]
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total
1	12.196	46.909	46.909	12.196	46.909	46.909	6.294
2	2.079	7.996	54.905	2.079	7.996	54.905	3.390
3	1.730	6.654	61.559	1.730	6.654	61.559	8.064
4	1.440	5.538	67.096	1.440	5.538	67.096	7.925
5	.957	3.679	70.775	.957	3.679	70.775	5.342
6	.881	3.388	74.163	.881	3.388	74.163	7.109
7	.664	2.553	76.717	.664	2.553	76.717	7.012
8	.589	2.265	78.981				
9	.525	2.018	80.999				
10	.497	1.910	82.910				
11	.484	1.861	84.771				
12	.429	1.651	86.421				
13	.399	1.536	87.957				
14	.361	1.388	89.346				
15	.326	1.253	90.598				
16	.295	1.136	91.735				
17	.281	1.081	92.816				
18	.268	1.032	93.848				
19	.260	1.002	94.849				
20	.239	.920	95.769				
21	.230	.883	96.652				
22	.204	.785	97.436				
23	.192	.739	98.175				
24	.175	.671	98.847				
25	.168	.646	99.492				
26	.132	.508	100.000				

6.4.6 Analysis of the Structural Equation Model

Inferring from the re-designed ABLA framework below in Figure 6.2 of the study, the researcher took interest to ascertain all the factors that enhance quality teaching and learning and also provide learning gains of the activity-based computer programming instructional approach within the blended learning environment. All the hypotheses were defined to ascertain the relationships and effects of the defined variables on students' learning experiences and learning gains in computer programming class among the three HEIs.



Abbreviations

AU- Activity-based Approach Used
SA- Social Activities
CA- Cognitive Activities
LG – Learning Gains
IA- Instructional Activities
SD – Skills Development
A&FB – Assessment and Feedback Activities

Figure 6.2 ABLA Framework with the Hypotheses of the Study

Hence, the researcher presents the main findings generated from the structural equation model after receiving new data from the students after the second semester of the AR. The findings that emanated from each research objective and hypotheses have been presented in the subsequent sections. Figure 6.3 demonstrates the confirmatory factor analysis developed from the study that constitutes all the key constructs used to measure the effect of activity-based computer programming instruction within a blended learning environment on students' Learning Gains, Instructional Activities, Social Activities, Cognitive Activities, Activity-based Learning Used and Assessment and Feedback activities.

6.4.7 Confirmatory Factor Analysis (CFA)

In the studies of Hair, Black, Babian and Anderson (2014), CFA is an approach to conduct factor analysis, mostly associated with SEM with the purpose of evaluating the magnitude to which a pre-defined structure fits the data. The analysis of the confirmatory factor was performed on the following factors:

- Activity-based Approach Used (AU)
- Social Activities (SA)
- Cognitive Activities (CA)
- Learning Gains (LG)
- Instructional Activities (IA)
- Skills Development (SD)
- Feedback Activities (FA)

Primarily, an assessment of the model fit for the confirmatory factor analysis was conducted based on the fit criteria as defined by Hu and Bentler (2006). Afterwards, the confirmatory factor analysis measure was presented.

Confirmatory Factor Analysis (CFA) enabled the researcher to test the proposed conceptual theory (Williams, Onsman and Brown, 2010, p. 3). Arbuckle (2013) argues that, factor loadings with critical ratios (CR) above 1.96 are significant at the 0.5 level and show a reasonable fit to data. In other words, for a relationship to be significant, the t-value (CR) should be more than absolute 1.96. The CFA test for this study was done in AMOS 26 and the t-values obtained for this study were more than 1.96 which went to indicate that they were significant and also showed a reasonable statistical acceptance.

6.4.8 Factor Loadings and Critical Ratios

Furthermore, a confirmatory factor analysis was run in AMOS 26.0, using the maximum likelihood estimation. This was to confirm the components or variables concluded after the exploratory factor analysis. According to Arbuckle (2013), factor loadings with critical ratios (CR) above 1.96 are significant at the 0.5 level and show a reasonable fit to data. The CFA indicated that the critical ratios were significant because they were all above 1.96. Results from the CFA, showcasing the factor loadings and their corresponding ratios are presented in Figure 6.3.

6.4.9 Structural Model for Confirmatory Factor Analysis

From the Figure 6.3 below, it is noted that the factors loaded well on each construct. All factors' loading was above 0.2.

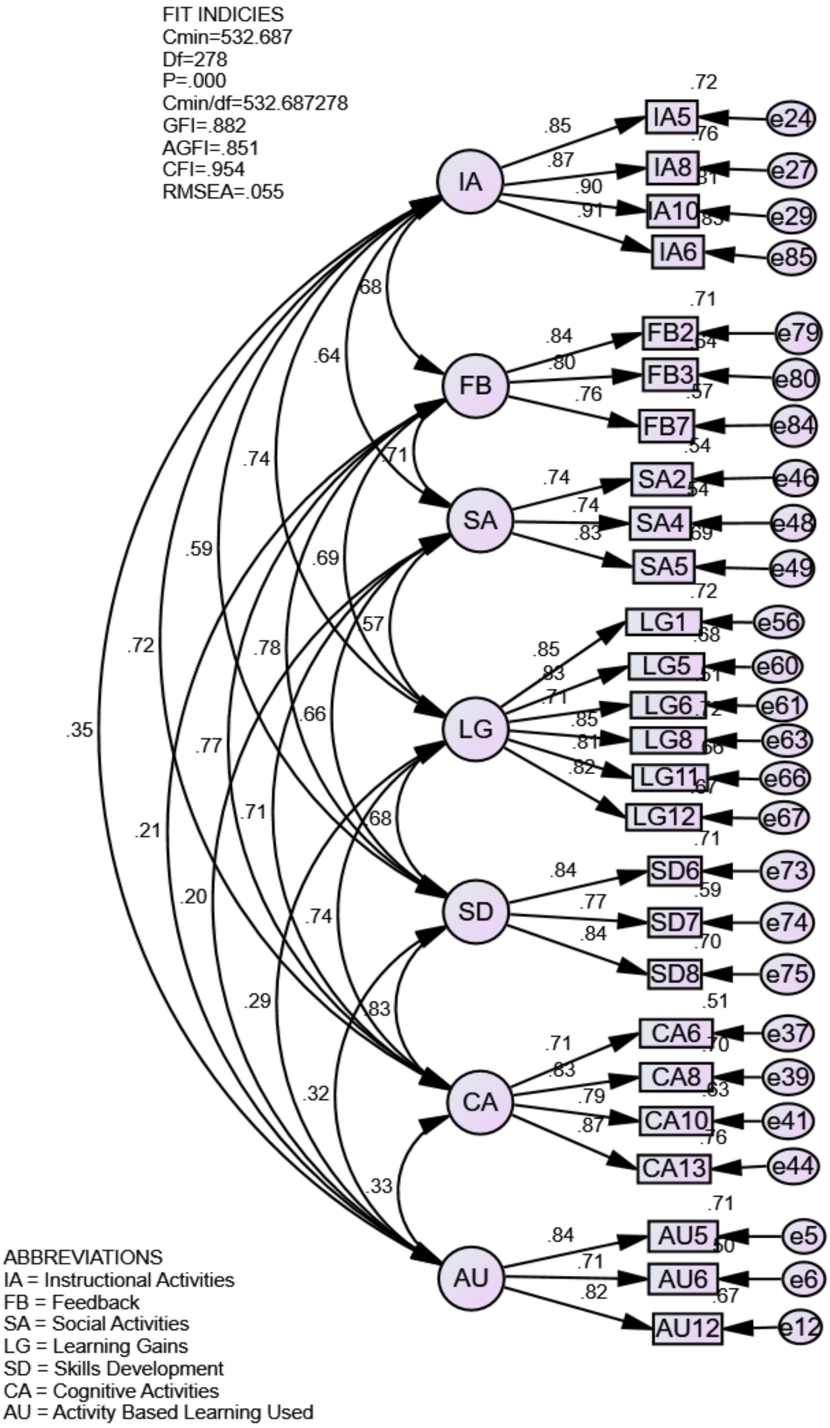


Figure 6.3 Structural Model for Confirmatory Factor Analysis

6.4.10 Common Method Bias

Common method bias is a potential problem in behavioural research. It is arguably one of the main sources of measurement bias that threatens validity (Lindell and Whitney, 2001). According to Nunnally (1978), the error is of two types, the systematic and random error. More seriously, systematic error consistently provides an alternative explanation for the observed relationship between measures of different constructs (Podsakoff, MacKenzie, and Lee, 2003).

Common method bias is evident when a single factor explains a majority of the data due to external factors. To check for this, the Herman's single factor test was conducted. The Herman's test requires that a single unrotated factor solution is factor analysed to determine if a single factor explains the majority of the variance in the model. In the study, CMB did not exist since the single factor accounted for 46.909% which was less than 50% as demonstrated in Table 6.7.

Table 6.7: Common Method Bias

Total Variance Explained						
Component	**Initial Eigen Values**			**Extraction Sums of Squared Loadings**		
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	12.196	46.909	46.909	12.196	46.909	46.909
2	2.079	7.996	54.905			
3	1.730	6.654	61.559			
4	1.440	5.538	67.096			
5	.957	3.679	70.775			
6	.881	3.388	74.163			
7	.664	2.553	76.717			
8	.589	2.265	78.981			
9	.525	2.018	80.999			
10	.497	1.910	82.910			
11	.484	1.861	84.771			
12	.429	1.651	86.421			
13	.399	1.536	87.957			

14	.361	1.388	89.346			
15	.326	1.253	90.598			
16	.295	1.136	91.735			
17	.281	1.081	92.816			
18	.268	1.032	93.848			
19	.260	1.002	94.849			
20	.239	.920	95.769			
21	.230	.883	96.652			
22	.204	.785	97.436			
23	.192	.739	98.175			
24	.175	.671	98.847			
25	.168	.646	99.492			
26	.132	.508	100.000			
Extraction Method: Principal Component Analysis.						

6.4.11 Model Fit for Confirmatory Factor Analysis

In determining how best the model fits the data, some generated fit indices after the CFA were observed. Researchers use numerous goodness-of-fit indicators to assess a model. Some common fit indexes are the Normed Fit Index (NFI), Non-Normed Fit Index (NNFI, also known as TLI) and Incremental Fit Index (IFI). The others are Comparative Fit Index (CFI), and Root Mean Square Error of Approximation (RMSEA).

The results obtained, threshold and interpretation which were automatically generated in AMOS 26 through Gaskin and Lim's (2016) plugin have been presented in the Table 6.8.

Table 6.8 Model Fit Information for CFA

Measure	**Estimate**	**Threshold**	**Interpretation**
CMIN	532.687	--	--
DF	278	--	--
CMIN/DF	1.916	Between 1 and 3	Excellent

CFI	0.954	>0.95	Excellent
SRMR	0.041	<0.08	Excellent
RMSEA	0.055	<0.06	Excellent
PClose	0.106	>0.05	Excellent

The popularity of fit-index research can be seen by the number of indexes that exist (Hu and Bentler, 1995). The indices obtained for this study included the following: chi square of 532.687, p= 0.000, a chi sq/df of 1.916, Comparative Fit Index (CFI) of 0.954, Adjusted Goodness-of-Fit Index (AGFI) of 0.916. The rest were Root Mean Square Error of Approximation (RMSEA) of 0.055, Root Mean Square Residual (RMR) of 0.47 and p of Close fit (PCLOSE) of 0.106 amongst others. Since all the indices obtained for this study are within the standardized ranges, it may be argued that the model fits the data collected from the respondents for this study. The cut off points, according to Hu and Bentler (2006), are presented in Table 6.9.

6.4.12 Cut-off Criteria

The cut-off criteria present the various cut-off points for the model fit indices and their corresponding interpretations as presented in Table 6.6. As may be observed in the table, according to Hu and Bentler (2006), the Cmin/Df should be less than 5, the Comparative fit index should be above .90, the Standardised root mean square residual should not be above 0.08 and the PClose should be more than 0.01. In the case of this study it was observed from the data that the CMI, CFI, SRMR, RMSEA and the PClose were excellent and were within the standardize rages per the cut-off criteria.

Table 6.9: Cut-off Criteria

Measure	Terrible	Acceptable	Excellent
CMIN/DF	> 5	> 3	> 1
CFI	<0.90	<0.95	>0.95
SRMR	>0.10	>0.08	<0.08
RMSEA	>0.08	>0.06	<0.06
PClose	<0.01	<0.05	>0.05

6.4.13 Confirmatory Factor Analysis Measures

The data outlining the factor loadings and the p-values are presented in Table 6.10. All factors were above 0.07. Items that recorded lower factor loadings were removed from the model to enhance the reliability of the measures.

Table 6.10 CFA Measures

Variables	**Path**		**Factor Loading**	**S.E.**	**C.R. (t-value)**	**P**
AU12	<---	AU	0.821			
AU5	<---	AU	0.840	0.073	13.405	***
AU6	<---	AU	0.707	0.082	12.073	***
CA10	<---	CA	0.794	0.096	13.132	***
CA13	<---	CA	0.869	0.106	14.321	***
CA6	<---	CA	0.713			
CA8	<---	CA	0.834	0.097	13.776	***
FB2	<---	FB	0.842			
FB3	<---	FB	0.798	0.056	15.491	***
FB7	<---	FB	0.757	0.061	14.447	***
IA10	<---	IA	0.901	0.054	20.800	***
IA5	<---	IA	0.846			
IA6	<---	IA	0.913	0.052	21.292	***
IA8	<---	IA	0.875	0.054	19.724	***
LG1	<---	LG	0.848			
LG11	<---	LG	0.813	0.052	17.302	***
LG12	<---	LG	0.817	0.057	17.431	***
LG5	<---	LG	0.826	0.056	17.762	***
LG6	<---	LG	0.714	0.063	14.232	***
LG8	<---	LG	0.846	0.052	18.456	***
SA2	<---	SA	0.737			
SA4	<---	SA	0.738	0.078	11.754	***
SA5	<---	SA	0.828	0.08	12.848	***
SD6	<---	SD	0.844	0.056	16.850	***
SD7	<---	SD	0.768	0.062	14.863	***
SD8	<---	SD	0.835			

6.4.14 Validity Analysis

The two types of validity analyses measured in this study were the convergent and discriminant validity. Also, Construct Reliability (CR) was measured to assess the reliability of the constructs.

Convergent validity refers to the degree to which two measures of construct that theoretically should be related are in fact related. To confirm convergent validity, the Average Variance Extracted (AVE) must be considered and the values for AVE must be AVE > 0.5, respectively (Hair, Black, Babin and Anderson, 2010). All the AVE figures obtained for this study fulfilled these requirements and as a result, it can be said that the study has convergent validity. Table 6.11 presents the outcome for the Convergent and Discriminant Validity.

Table 6.11 Convergent and Discriminant Validity

	CR	**AVE**	**MSV**	**Max R(H)**	**AU**	**IA**	**SA**	**LG**	**SD**	**FB**	**CA**
AU	0.833	0.627	0.119	0.845	0.792						
IA	0.949	0.790	0.546	0.952	0.345	0.889					
SA	0.812	0.592	0.508	0.820	0.204	0.628	0.769				
LG	0.920	0.659	0.546	0.924	0.293	0.739	0.569	0.812			
SD	0.857	0.667	0.691	0.862	0.319	0.587	0.660	0.682	0.817		
FB	0.842	0.640	0.608	0.847	0.207	0.687	0.710	0.690	0.780	0.800	
CA	0.879	0.647	0.691	0.890	0.330	0.717	0.713	0.736	0.831	0.774	0.804
Abbreviation: AU = Activity Based Learning Used, IA = Instructional Activities, SA = Social Activities, LG = Learning Gains, SD = Skills Development, FB = Feedback, CA = Cognitive Activities											

On the other hand, discriminant validity refers to the degree to which two measures of construct that theoretically should not be related are in fact unrelated. For discriminant validity to exist, Maximum Shared Square Variance (MSV) should be less than AVE or MSV should be less than the AVE squared. The figures obtained for this study indicated that, discriminant validity existed for all constructs with the exception of skills development and cognitive activities. That notwithstanding, the figures are just below the threshold and all other indices are favourable. For theoretical relevance, all seven constructs were maintained.

6.4.15 Assessing the Effects

The study established that a significant relationship was observed between the cognitive activities and activity-based learning used (standardized coefficient = 0.928, $p<0.01$). Also, a significant relationship was found between social activities and activity-based learning used (standardized coefficient = 0.785, $p<0.01$). Instructional activities are significantly affected by the activity-based learning used (standardized coefficient = 0.764, $p<0.01$). The main structural equation model showing relations among the constructs is shown in Figure 6.4 below.

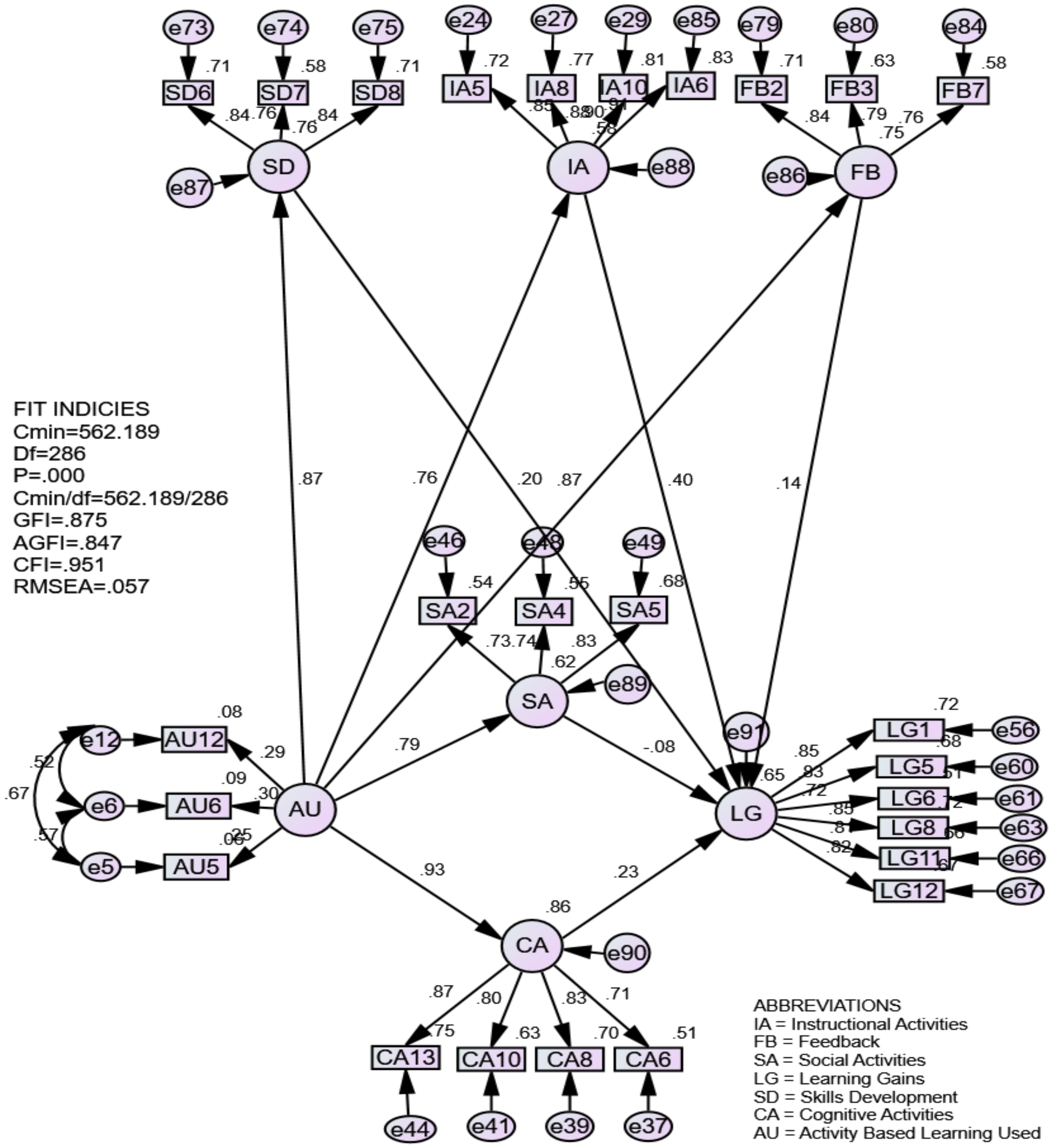


Figure 6.4: Main Structural Equation Model

In fact, from Table 6.12, it is noted that all relationships are significant at $p<0.001$ with the exception of the relationship between social activities and learning gains. Most importantly, the sequence of the standardized regression weights shows the various extents of impact of the variables on each other.

Table 6.12: Assessing the Effects

	Path		Regression Weight	S.E.	C.R.	P
CA	<---	AU	0.928	0.382	4.700	***
SA	<---	AU	0.785	0.362	4.595	***
IA	<---	AU	0.764	0.430	4.705	***
SD	<---	AU	0.871	0.472	4.771	***
FB	<---	AU	0.866	0.465	4.766	***
LG	<---	CA	0.233	0.149	2.109	0.035
LG	<---	SA	-0.083	0.093	-1.093	0.274
LG	<---	IA	0.405	0.065	6.099	***
Indirect Path						
LG	<---	AU	0.750	0.028	3.170	0.021

Abbreviations: AU = Activity Based Learning Used, IA = Instructional Activities, SA = Social Activities, LG = Learning Gains, SD = Skills Development, FB = Feedback, CA = Cognitive Activities

There is a significant positive relationship between skills development and activity-based learning used (standardized coefficient = 0. 871, p<0.01). Activity-based learning used has a significant effect on feedback (standardized coefficient = 0.886, p<0.01). Cognitive activities have a positive impact on learning gains (standardized coefficient = 0.233, p<0.05). The relationship between social activities and learning goals was insignificant (standardized coefficient = -0.083, p = 0.274). Finally, instructional activities have a positive effect on learning gains (standardized coefficient = 0.405, p<0.01). Activity based learning used has a significant positive effect on learning gains.

6.4.16 Structural Model of the Combined Effect

The combined effect of all activities on learning gains was assessed in the model indicated in Figure 6.5.

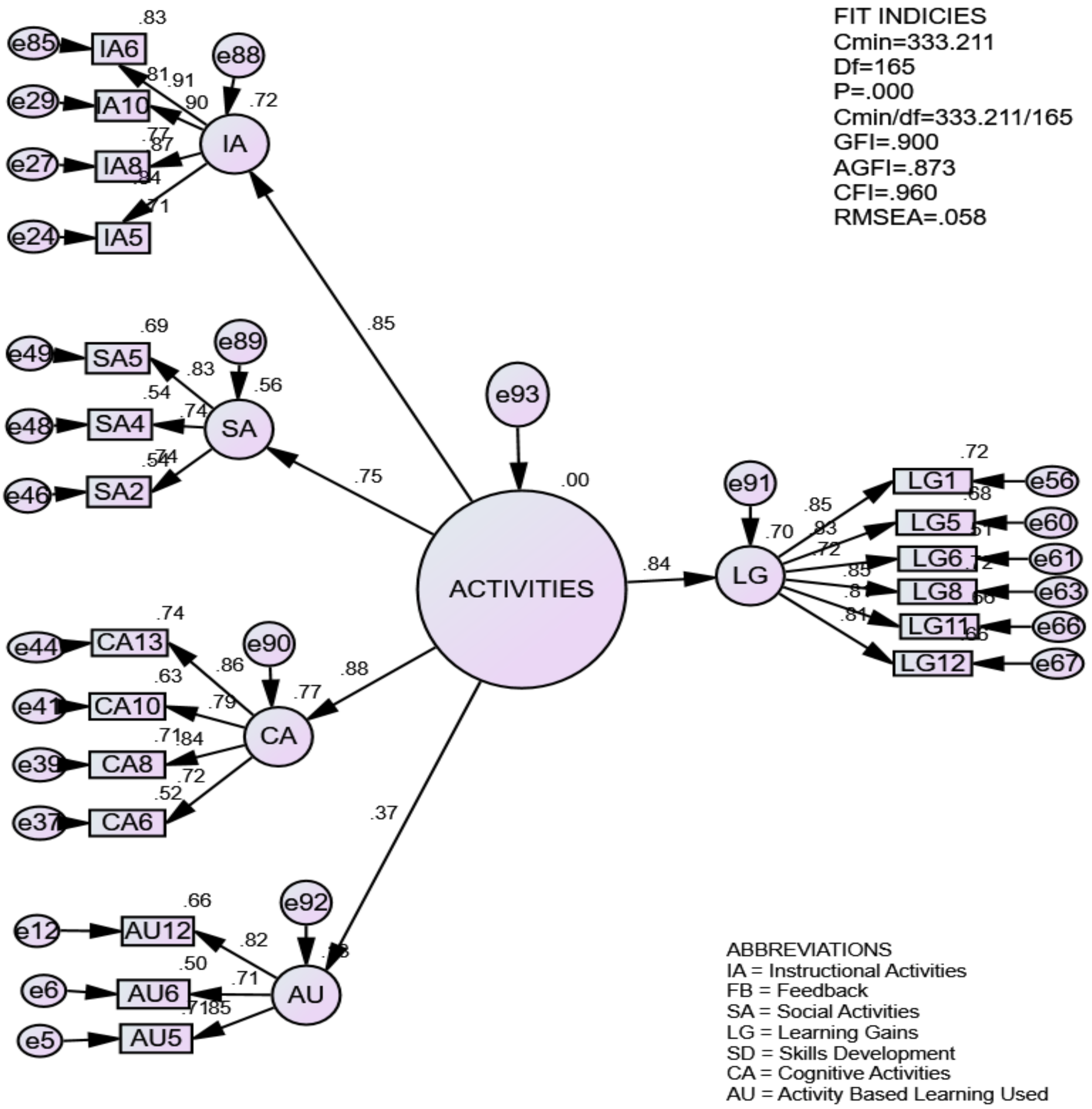


Figure 6.5: Structural Model of Combined Effect

6.4.16.1 Model Fit Indices for Combined Effect

From Table 6.13 below, the indices obtained for this study are presented which include a chi square of 333.211, p= 0.000, a chi Sq/df of 2.019, Comparative Fit Index (CFI) of 0.960, and Adjusted Goodness-of-Fit Index (AGFI) of 0.915. The rest are Root Mean Square Error of Approximation (RMSEA) of 0.058, Root Mean Square Residual (RMR) of 0.58 and p of Close fit (PCLOSE) of 0.064 amongst others. Since all the indices obtained for this study are within the standardized ranges, it may be argued that the model fits the data collected from the respondents for this study.

Table 6.13: Model Fit Indices for Combined Effect

Measure	Estimate	Threshold	Interpretation
CMIN	333.211	--	--
DF	165	--	--
CMIN/DF	2.019	Between 1 and 3	Excellent
CFI	0.960	>0.95	Excellent
SRMR	0.046	<0.08	Excellent
RMSEA	0.058	<0.06	Excellent
PClose	0.064	>0.05	Excellent

In Table 6.13, it is found that there is a significant positive effect of the combined effect of the four activities (Instructional activities, Social activities, Cognitive activities, Activity-based learning used) on learning Goals (standardized coefficient = 0.835, p<0.01). It is worth noting that altogether, the activities have a significant impact on the learning gains but from the previous model (see Figure 6.4), the social activities alone did not have a significant impact on the learning gains. In other words, it is worth to know that being socially active in class does not necessarily guarantee learning gain. The path assessment output is presented in Table 6.14 below.

Table 6.14: Path Analysis for Combined Effect

Paths			Regression Weight	S.E.	C.R.	P
IA	<---	ACTIVITIES	0.850			
SA	<---	ACTIVITIES	0.747	0.074	9.579	***
CA	<---	ACTIVITIES	0.880	0.071	10.836	***
AU	<---	ACTIVITIES	0.367	0.086	5.339	***
LG	<---	ACTIVITIES	0.835	0.08	12.076	***

After the analysis of the quantity data of the ABLA framework using SEM, the researcher further performed analysis of the data derived from the qualitative data on the ABLA framework.

6.5 Qualitative Analysis of the Study During the Third Semester

In the previous semesters, the researcher ascertained the activity-based approaches used in computer programming instructions and developed a curriculum that aids its delivery. To establish a robust approach that will enhance an effective teaching and learning of computer programming, the researcher during the third semester analysed and reflected on the qualitative data on the ABLA framework. It is in this regard that the researcher conducted semi-structured interviews and focused group discussions to ascertain views of the students and faculty on the use of the ABLA framework used in teaching computer programming within a blended learning environment. The interview guide used to collect the qualitative data on the ABLA framework can be seen in Appendix F.

6.5.1 Profile of the Participants (Students)

A total of 18 students responded to the qualitative data gathering instrument via a self-administered interview and focused group discussions. A total of 9 students were involved in three different focused group discussions among the three HEIs of three students each. This was done in a face-to-face meeting before COVID-19 (from November, 2019 to February, 2020) while 3 students from each of the three HEIs totalling 9 students were interviewed via Zoom, (an online communication tool) as demonstrated in Table 6.15. For confidentiality purposes, the interviewees are represented as Student A, Student B, Student C with their Institutions as HEI-1, HEI-2 and HEI-3. The interviewees were predominately Information Technology and Computer Science students. With gender, there was a gender balance with respect to each institution. Predominately, the levels of students interviewed were in levels 100, 200, and 300.

Table 6.15: Demographics of Interviewees (Students)

Interviewee	Gender	Institution	Level	Program
Student A HEI-1	Female	HEI-1	100	Information Technology
Student B HEI-1	Male	HEI-1	300	Computer Science
Student C HEI-1	Male	HEI-1	200	Computer Engineering
Student A HEI-2	Male	HEI-2	100	Information Technology
Student B HEI-2	Female	HEI-2	200	Information Technology
Student C HEI-2	Male	HEI-2	400	Computer Science
Student A HEI-3	Male	HEI-3	300	Computer Science
Student B HEI-3	Female	HEI-3	100	Information Technology
Student C HEI-3	Male	HEI-3	200	Information Technology
Total Interview Responses = 9				

6.5.2 Interview Responses (Students)

As indicated already, the study followed a sequential mixed-method approach of the study. The responses from the interview were captured and thematic analysis was used to present the students' learning experiences of the activity-based computer programming instruction within a blended learning environment. The interview guide is attached as Appendix F.

Yin (2013) contended that the use of computer-based application is very important for analyzing data. Nevertheless, Yin viewed that qualitative data analysis should be conducted using human thoughtfulness and collaboration with the respondents. Hence, the researcher used a rigorous cognitive approach to analyze the data and interpret it using thematic analysis as recommended by Taylor-Powell and Renner (2003).

Taylor-Powell and Renner suggested the following phases for performing qualitative analysis:

1. Familiarization of the data
2. Concentrating the analysis
3. Grouping the information
4. Establishing patterns and categories
5. Interpreting the analyzed data

Adhering to the approach of Taylor-Powell and Renner (2003) on the data, statements, contents, themes and sub themes were classified and presented. Hence, the data were categorized in three different major themes. These were the following:

1. The students' learning experience and perception of the ABLA framework.
2. The students' learning experience on computer programming using the ABLA Model.
3. The students' experiences of the ABLA framework within the blended learning environments and provisioning of learner support.

6.5.3 The Students' Learning Experience and Perception of the ABLA Framework Used in Computer Programming Instruction

To ascertain the students' learning experience and perception of the ABLA framework used in the computer programming instruction, the research conducted an interview during the third semester of the AR using an interview guide as seen in Appendix F. Firstly, the students were asked to provide their learning experiences and perceptions on the activity-based approach used in teaching computer programming. The researcher wanted to find out how most students perceived the activity-based approach per the ABLA framework and consequently how the approach affects their learning journey in computer programming classes.

The findings from the study indicated that most students are appreciative towards the use of the activity-based learning approach in teaching and learning computer programming. However, some students expressed interest for improvement and more engagements of computer programming assignments and activities.

The comments raised by the some of the interviewees are indicated below:

Interviewee Student A HEI-1

"Yes, my instructor uses the activity-based approach in our class. I say so because he mostly engages us with series of group discussions, projects, case studies, and practical sessions".

Interviewee Student B HEI-1

"I am personally fine with the way my lecturer teaches programming, it has benefited me a lot, I can now code most of the assignments with less support from my friends"

Interviewee Student C HEI-3

"Some of the learning gains that benefited me most in my programming class are, I can now understand the C++ syntax very well. Initially, I found it very difficult to understand conditional statements, loops, arrays and functions but now I can write codes using if else condition, for loops and functions".

Interviewee Student A HEI-2

"On the social activities, our lecturer sometimes groups us into groups of three or five members and give us long case studies and projects, after he will tell us to submit online via the e-learning platform. When we meet in class he will then call us or anyone in our group to make presentations in class. Sometimes, he also tells us to respond to post on the forum to discuss."

Interviewee Student B HEI-2

"I had in mind before applying for this course that I was going to be a professional in programming, but looking at the performance of some programming lecturers, they have not been able to help us practically, I just feel like giving up and moving to a different and more practical oriented institution. Also, my instructor barely does not use the activity-based approach frequently."

Interviewee Student C HEI-3

"The activity-based approach employed is not standard to compel us to have full focus in programming course. Creating good learning environment in class with well experienced lecturers will serve as a complement to achieving the goal of activity-based approach."

Interviewee Student C HEI-1

"On the assessment, my lecturer mostly gives as case studies, projects and quizzes. Yes, he assesses us almost at the end of each topic. Last semester we wrote the final exams physically

on campus. But the continuous assessments were all in class activities, group work, case studies, and assignments"

It was noted from the students' comments that most of them expressed missed opinions to the responses on their experiences of the activity-based approach used in their respective institution. It was established that the approach was helpful to some of the students in the HEIs, example is in the case of Interviewee Student A HEI-1, Student B HEI-2, Student C HEI-3, and Student C HEI-1, etc.

Generally, most of the students expressed a significant impact of the activity-based approach used in their class on their learning gains. Most of the students interviewed such as Student A HEI-1, Student B HEI-1, Student C HEI-3 and Student A HEI-2 affirmed that activity-based approach has enhanced their cognitive (comprehension of programming syntax and logics) and social collaborative engagements which connects with the study of Cripps, Jacobs, and MacCallum (2020).

Of course, the researcher believes that every student has different learning styles, pace and needs, hence the activity-based approach did not meet some of the students' learning expectations in computer programming. This implies that the instructors will have to improve on areas that demonstrated less desired approach of using the activity-based approach to teach computer programming as indicated by Interviewee Student C HEI-3 and Interviewee Student B HEI-2.

6.5.4 Qualitative Analysis of Instructors on the ABLA Framework for Instructing Computer Programming in HEIs.

In this section, the researcher gathered qualitative responses on the effectiveness and relevance of the ABLA framework that supports activity-based computer programming instruction (See Appendix G for the instruments used to collect data from the instructors). The data was collected and transcribed to gather different meanings from the instructors. Thematic analysis as guided by Taylor-Powell and Renner (2003) and Burnard, (1991) was employed using Atlas Ti. The data was thematically coded by reading the transcripts and defining tentative categories, themes and counts on the occurrences. The data was collected through face-to-face interview meetings and online interviews via Zoom after COVID-19 to ascertain instructors' opinions on the framework for activity-based programming instruction.

6.5.4.1 Demographics of Instructors

A total of six instructors responded to the qualitative data gathering instrument through an interview for the three HEIs. It means there were two interviewees from each of the three institutions as indicated in Table 6.16. In view of confidentiality, the interviewees are represented as Instructor A and Instructor B with their Institutions as HEI-1, HEI-2 and HEI-3. The interviewees were all programming lecturers. Only one of them has 8 years of teaching experience; the rest have between 9 and 14 years of teaching experience in computer programming. As regards gender, there was no female among the instructors in all the three institutions. Predominately, the number of years of delivery via a blended learning approach were noted to be very encouraging with at least 3 years of experience while the other instructors indicated 8, 4, 6, 7, and 5 years respectively.

The programming courses taught included high-level programming with C++, Java Programming, Advanced Programming with C#.net, VB.net and web programming with PhP programming languages.

Table 6.16: Demographics of the of Instructors

Interviewee	Gender	Institution	No. of Years Teaching	No. of Years in using Blended Learning	Programming Courses Taught
Instructor A HEI-1	Male	HEI-1	10	8	High-level Language (C++)
Instructor B HEI-1	Male	HEI-1	14	4	Java Programming
Instructor A HEI-2	Male	HEI-2	12	6	Programming I (C++)
Instructor B HEI-2	Male	HEI-2	8	3	Advanced Programming (C#.net and VB.net)
Instructor A HEI-3	Male	HEI-3	9	7	Programming with C++
Instructor B HEI-3	Male	HEI-3	11	5	Web Programming (PhP)
Total Interview Responses = 6					

6.5.5 Thematic Analysis of Instructors' Experiences on ABLA Framework for Computer Programming Instruction.

After coding and thematically organizing the transcripts, occurring responses that were related and similar were put into different categories and themes as seen in Table 6.17.

Table 6.17: Thematic Analysis of Instructors' Experiences on the ABLA Framework for Programming Instruction

Category	Theme	Occurrences
Activity-based Experience within Blended Learning Environment	The use of activity-based approach	5
	The activity-based approach was effective	6
	The activity-based approach used within the BL environment enhanced delivery	4
	The activity-based programming platform (ABPI) was very user friendly and was appreciated by the students	5
	The LMS supported the activity-based approach used in teaching programming where distance was not a barrier.	4
Computer Programming Teaching and Learning Experience	The programming concepts have always been difficult with students	6
	The practical activities were appreciated by most students	5
	Group activities in class were good	6
	Feedbacks on assignments affected the students' programming skills	4
	Problem-based activities were seen to enhance students programming experiences.	5
	The video resources used enhanced students cognitive and practical skills to programme.	6
Satisfaction	Satisfied with activity-based approach used	4
	The activity-based approach has seen a significant improvement since 50-60% of the students in the class have now begun to develop programmes.	5

	Internet connectivity is largely seen as a challenge for online engagements	6
	The LMS support team were very helpful	5
	ABPI instructional platform is easy to use	4
Curriculum Mediations	The curriculum developed was suitable for computer science foundational courses	5
	The learning outcomes were lucid	6
	The adapted curriculum/syllabus/course outline support the activity-based approach in teaching programming in my university is effective	5
	The pedagogic approach using the activity-based approach should be encouraged in all computing science courses.	5
	The assessment methods should be more formative rather than summative and should carry greater percentage of the total assessment to foster the activity-based approach.	4

The analysis of each category and theme derived from the thematic analysis is discussed in the subsequent sections.

6.5.5.1 Activity-Based Teaching Experience within Blended Learning Environment

The researcher's first assessment of the thematic and content analysis gathered from the qualitative data was the instructors' activity-based teaching experience within the blended learning environment. It was generally established that five out of the six lecturers interviewed used the activity-based approach to teach programming. On the instructors' experiences on the activity-based approach used in their programming class, all the six lecturers affirmed that the approach was very effective and thus, it has helped them to develop/improve their teaching skills in programming.

Again, four of the instructors indicated that the activity-based approach used within the BL environment enhanced their delivery skills. That is to say, the blend of both the face-to-face and online delivery enhance seamless engagement with the students and lecturers. More so, six instructors affirmed that the LMS supported the activity-based approach used in teaching programming where distance was not a barrier. Thus, students can participate in quizzes, assignment submissions, contribute to forum discussions and many more via the LMS platform.

6.5.5.2 Computer Programming Teaching and Learning Experience

The second category sought to establish computer programming teaching and learning experiences among the instructors following the ABLA framework. It was lucidly noted by all the six instructors that most students have challenges in programming language courses. This is an affirmation of the already defined challenges most of the students asserted in the quantitative data as indicated in Table 5.8 in the previous chapter. Also, five lecturers indicated that the activity-based approach enhances the practical engagements of the students.

As indicated in the previous chapter, students attributed group activities in class as a good practice that fostered their programming experience. Their six lecturers, also responded in the affirmative that group activities assigned to students for programming activities in class and outside the classroom supported their cognitive development and programming skills (i.e. students are able to learn from their peers). Four instructors reported constant feedbacks on assessments affected their programming skills positively.

Again, it was discovered from the interviews with lecturers that problem-based activities were seen to enhance students programming experiences with five instructors responding in affirmative. A further probing on why the problem-based activities enhances the students programming activities discovered that the lecturers mostly give them practical problems that are applicable to the day-to-day business activities. Hence, most students are enthused to work on such problems and challenges them to program. It was also discovered that some Faculty members who teach programming employ the use of the flip method of delivery where they make video resources available to the students to watch before they come to class for further discussions. Interestingly, all the six instructors responded in affirmative that the video resources used enhanced students' cognitive and practical skills to code.

6.5.5.3 Satisfaction

Another category that emanated from the thematic analysis is the evaluation of the satisfaction level of instructors with the use of the ABLA framework for teaching and learning of computer programming within the blended learning environment used. It was generally observed that four instructors were satisfied with activity-based approach used in programming instruction. Captivatingly, five lecturers affirmed that the ABLA framework has seen a significant improvement in their class since 50-60% of their students have now begun to develop programming. Contrary to the positive responses from the instructors, all the instructors (6) interviewed alluded to the fact that internet connectivity was largely seen as a challenge for online engagements for both students and lecturers.

Another emerging discovery from the qualitative responses was the provision of learner support as posited by Ahsan and Mullick (2013). The researcher also found out from the lecturers that the LMS support team were very helpful and provided seamless teaching and learning experience to both students and instructors.

In conclusion, the findings from both the qualitative and quantitative data has shed light on the positive impact of the ABLA framework on the students learning gains. As a result of this findings, the researcher further performed a reflection on the findings of the ABLA framework to ascertain its effective usefulness for the teaching and learning of computer programming within a blended learning environment. This is discussed in the subsequent sections.

6.6 Reflections and Re-designing of the ABLA Framework for Computer Programming Instruction

Following the action research approach adapted for the study, the researcher at the 6th stage (i.e. Semester 3), conducted a personal reflection on the findings and the design of the ABLA framework. The researcher sought inputs from his supervisors, the Head of Department (HOD) of the Computer Science division of the researcher's institution and the six instructors in the three HEIs to evaluate the findings derived from both the quantitative and qualitative data of the study throughout the three semesters research periods. The Feedback from the instructors and the HOD were very encouraging and thus, the instruments used was good and relevant for an effective teaching and learning of computer programming in HEIs within a blended learning environment.

The researcher also noticed from both the qualitative and quantitative data that the activity-based approach affected the programming skills of the students positively. From a personal observation as an instructor and the feedback from colleagues in the three HEIs solidly affirmed that the activity-based approach for teaching and learning programming within the blended learning environment supported the students' cognitive and programming skills. Also, most of the students equally saw the activity-based approach as an effective approach of teaching and learning programming as seen in their responses in the interview conducted in section 6.5.5. Again, the researcher saw the need to equally use the activity-based pedagogic approach for teaching computing science courses to enhance the students' capabilities of solving real-world problems and consequently increasing their employability opportunities after they earn their degrees.

Finally, the researcher saw the need to redesign the ABLA framework considering the various constructs that supports teaching and learning of computer programming within a blended learning environment and also provided learning gains to the students as seen in Figure 6.6 below. It was found that there is a significant positive effect of the combined effect of the four activities (Instructional activities, Social activities, Cognitive activities, Activity-based learning used) on learning Goals (standardized coefficient = 0.835, $p<0.01$). Upon reflecting on these findings, the combined activities have a significant impact on the learning gains but from the previous model (see Figure 6.4), the social activities alone did not have a significant impact on the learning gains hence the redesign of the ABLA framework as seen in Figure 6.6.

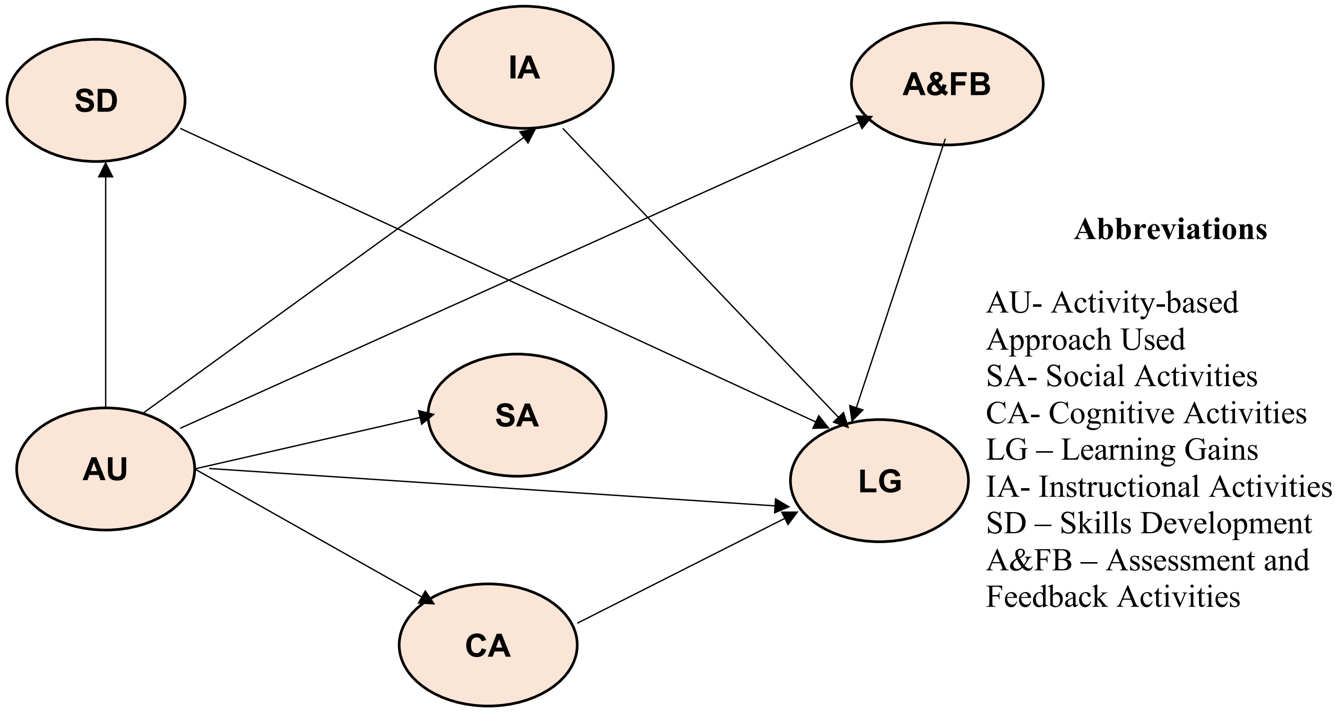


Figure 6.6 Re-design of the ABLA Framework

6.7 Reflections on the Students' Experiences of the Blended Learning (BL) Environments and Provisioning of Learner Support

The researcher performed a reflection on the BL environment as a vehicle to support the delivery of the activity-based approach used for the instruction of computer programming. Hence, the researcher asked the students on their experiences of the use of the blended learning environment and provisioning of a learner support as posited by Ahsan and Mullick (2013); Anderson and Elloumi (2004); and Dietinger (2003). It was discovered from the qualitative data that the students appreciated the blended learning environment for computer programming instruction. The BL environment provided to the students gave them an opportunity to learn using both face-to-face and virtual approaches.

The online component of the BL provided to the students a platform where most students contributed to threaded forum discussions, perform quizzes, and submit assignments online. The students and lecturers confirmed that the administrators and managers of the e-learning platform were seen as very supportive towards teaching and learning of computer programming using the BL approach. This was evident in their prompt responses to resolve issues on the LMS platforms and routine remote chatting with students when students encounter issues online. However, some of the students expressed concern with connectivity issues, especially when away from the learners' institutions. Below are some feedbacks and comments from the interviewee students on the BL environment and provisioning of learner support.

Interviewee Student A HEI-1
"The e-learning platform has been very consistent and is user friendly. The support from the TAs were good"

Interviewee Student A HEI-2
"I appreciate the support services available to support me online anytime I get issues accessing the platform. The staff are very supportive to my learning journey. However, we always find it difficult to contact our lecturer after lectures."

Interviewee Student C HEI-2
"I like the blended learning environment. I can sit at home and chat with my colleagues virtually on the LMS platform. I can also submit my assignments without traveling to campus. Honestly, the TA is doing even better than the lecturer in terms of teaching programming to my understanding."

Interviewee Student B HEI-1
"Internet connectivity has always been my challenge most times when I am not on campus. The school must do something about it. On the support services, my lecturer has created that opportunity for us to meet him on Tuesdays and Fridays."

Interviewee Student C HEI-1
"Data is too expensive. Most times I am unable to meet my deadlines because of connectivity issues. Apart from that the blended learning environment is good."

Interviewee Student B HEI-3
"The Online Support Staff prompt response to resolve issues on the LMS platforms is very lovely and prompt".

Interviewee Student B HEI-2

"I can sit home and chat with my peers, contact my lecturers and do quizzes with ease. The school should encourage this. The support from the TAs is just awesome."

It was discovered from the qualitative data that some students lacked support after their classroom activities on their computer programming journey. This worry, as expressed by some students, occurred to situations where some of the lecturers were adjunct faculty and does not mostly stay on campus after lectures. However, some lecturers were available to support the students beyond the synchronous face-to-face delivery for support.

Moreover, some of the HEIs have Tutors or Teaching Assistants (TA) that support the computer programming students after their main class sessions. Some students saw the Teaching Assistants' support as very useful and good as commented by Interviewee Student C HEI-2. Generally, the researcher believes and rates the learner support services employed among the three HEIs as very good and appreciable towards teaching and learning of computer programming.

6.8 Relevance of the third Semester Findings on the ABLA Framework Development

During the third semester of the semester AR, the researcher analysed the various constructs that contributes to the development of the ABLA framework. The findings established in the third semester aided the researcher to affirm the relevance of the constructs on the activity-based programming instruction per the conceptual framework used for the study. The findings also aided the researcher to ascertain the various effects of the constructs on the learning gains towards the ABLA framework development. The findings contributed about 80% of the action research towards the development of the ABLA framework. The progress view of the AR for the ABLA framework is represented in Figure 6.7 below.

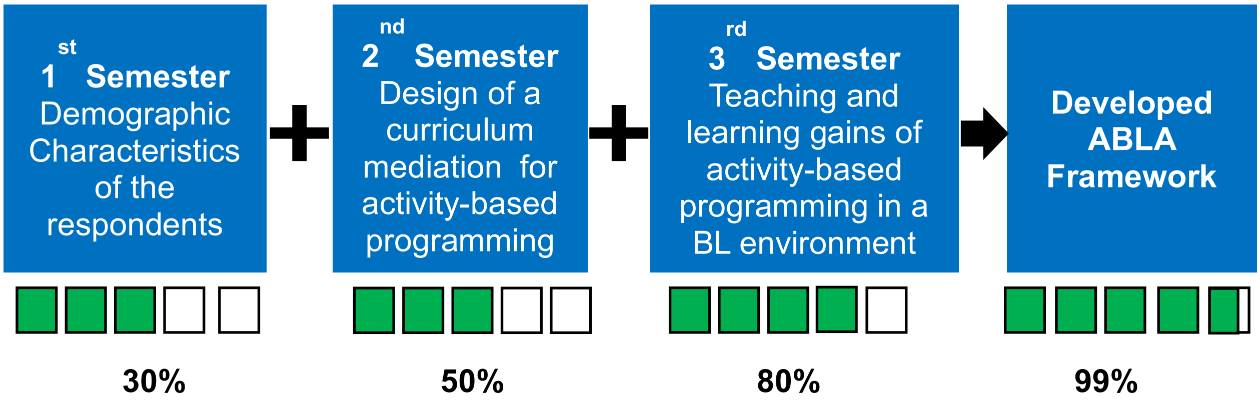


Figure 6.7 Progress View Towards ABLA Framework Development as at Semester 3

6.8 General Reflections of Data Analysis and Presentation of Results for First, Second, and Third Semesters of the AR Research

In chapters 5 and 6 of this study, the researcher has detailed and presented the data analysis and results obtained from the study from the very first semester to the third semester of the action research process. The analysis was organized and captured both qualitative and quantitative data. The data on the demographic characteristics of the students, an ANOVA test to measure the variances of the selected institutions and the types of activity-based approach used in computer programming instruction for the first semester was established. Most students attributed the lack of computer programming skills to the approach their lecturers use to teach them. Some also expressed concern that understanding programming syntax and semantics were seen as a challenge.

The curriculum mediation based on pedagogic approaches and technology mediation using ABPI for teaching computer programming was implemented in the second semester of the AR. The ABLA framework was developed and implemented in the third semester and on the third semester. The data collected in the third semester was preliminary analysed with SEM. Data screening were presented, and then, validation of measurement, exploratory factor analysis, confirmatory factor analysis, and the analyses of hypotheses were conducted. The ABLA framework was finally re-designed based on the findings derived from the structural equation model.

The chapter also analysed the qualitative data for both students and lecturers on their experiences on the ABLA framework. The qualitative aspect of the data analysis was done using thematic and content analysis. Hence, the qualitative data were categorized into three different major themes for the students. These were students' learning experience and perception of the activity-based approach used in computer programming instruction, the students' learning experience on computer programming and the students' experiences of the blended learning environments and provisioning of learner support. The findings from the perceptions and on the learning experiences of the ABLA framework for programming instruction was seen as a critical aspect of the students' learning journey that supports learning gains.

Finally, findings from the interview discussions with the six computer programming instructors on their experiences of the ABLA framework on their instructional approach in computer programming were ascertained and the conclusions proved very positive and relevant.

It was largely seen from the interview responses that teaching and learning of computer programming were seen as a major challenge among the students. The researcher finally performed a reflection on the findings from the ABLA framework for teaching and learning of computer programming within a blended learning environment.

6.9 Testing and Evaluation of ABLA Framework Per the Hypotheses

Hypothesis 1: Hypothesis 1 states that the activity-based teaching and learning of computer programming in HEIs within a blended learning environment include but not limited to case studies, quizzes, projects, group discussions and presentations, problem-based learning and concept mapping. The descriptive analysis proved that the current activity-based approach used among the HEIs are Case Studies, Quizzes, Projects, Group Discussions and Presentations, Educational Games, Videos, Debates, Problem-based Learning, Field Work, Concept Mapping and the use of TEL tools. The ANOVA test conducted shows that there were some significant differences between the approaches used among the three HEIs.

Hypothesis 2: This hypothesis states that the new developed curriculum used in teaching computer programming within a blended learning environment supports activity-based teaching and learning. The findings from the study affirms that the new developed curriculum supports activity-based computer programming instruction within a blended learning environment in Ghana.

Hypothesis 3: Hypothesis 3 states that activity-based learning approach of teaching computer programming within HEIs in Ghana has a significant positive effect on students' learning gains. It was shown in Table 5.19 that activity-based learning approach of teaching programming within HEIs in Ghana had a significant positive effect on the learning gains ($r=0.835$, $p<.000$). A unit increase in activity-based learning approach of teaching computer programming within HEIs results in 85% increase in the student learning outcomes.

Hypothesis 3.1: This hypothesis states that the activity-based learning approach of teaching computer programming within HEIs in Ghana has a significant positive effect on students' cognitive development. This hypothesis was supported with effect of $r=0.880$, $p<0.000$. a unit change in activity-based learning approach of teaching computer programming within HEIs in Ghana results in an 88% change in students' cognitive development.

Hypothesis 3.2: Hypothesis 3.2 states that the activity-based learning approach of teaching computer programming within HEIs in Ghana has a significant positive effect ($r=0.7847$, $p<0.000$) on students' social activities and learning. This was found to be significant. A unit increase in the activity-based approach of teaching computer programming will result in a 74.7% increase in a student's social activities.

Hypothesis 3.3: Hypothesis 3.3 which states that the activity-based learning approach of teaching computer programming within HEIs in Ghana has a significant positive effect on Instructional Activities was also accepted ($r=0.850$, $p<0.000$). This implies that a unit change in activity-based learning approach of teaching computer programming within HEIs in Ghana will result in an 85% increase in instructional activities.

Hypothesis 3.4: This hypothesis suggests that the activity-based learning approach of teaching computer programming within HEIs has a significant positive effect on Skills Development. This was also found to be significant ($r=0.871$, $p<0.000$). A unit change in activity-based learning approach of teaching computer programming within HEIs in Ghana will hence result in an 87.1% increase in skills development.

Hypothesis 3.5: This hypothesis was also found to be significant (r=0.866, p<0.000). Activity-based learning approach of teaching computer programming within HEIs in Ghana has a significant positive effect on students' feedback and assessments. This means that a unit increase in activity-based learning approach of teaching computer programming within HEIs in Ghana will result in an 86.6% increase in students' feedback and assessment.

Hypothesis 3.6: Finally, Hypothesis 3.6 which states that social activities have a significant positive effect on learning gains was unsupported (r=-0.083, p=0.274). Social activities employed in teaching computer programming within HEIs in Ghana has no significant positive effect on students' learning gains.

6.10 Conclusion Derived from the Evaluated Hypotheses

In an attempt to answer all the research questions, the researcher ascertained whether the hypotheses derived from the research questions were supported or not. Table 6.4 outlines decisions made on the hypothesized research questions.

Table 6.18 Conclusion Derived from the Research Hypotheses

Research Questions	**Hypotheses of the research**	**Decision**
What is the current activity-based learning approaches in computer programming instructions within a blended learning environment in HEIs.	H_1: The current activity-based teaching and learning of computer programming in HEIs within a blended learning environment are case studies, quizzes, projects, group discussions and presentations, problem-based learning and concept mapping.	Supported
What curriculum mediation based on pedagogic approaches could support activity-based learning for instructing computer programming in HEIs?	H_2: The developed curriculum used in teaching computer programming within a blended learning environment supports activity-based learning.	Supported

What is the learning and teaching gains of an activity-based learning approach in a blended learning environment for students and lecturers in HEIs?	H_3: The activity-based learning approach of teaching computer programming within HEIs has a significant positive effect on students learning gains.	Supported
	$H_{3.1}$: The activity-based learning approach of teaching computer programming within HEIs has a significant positive effect on students' cognitive development.	Supported
	$H_{3.2}$: The activity-based learning approach of teaching computer programming within HEIs has a significant positive effect on students' social activities and learning.	Supported
	$H_{3.3}$: The activity-based learning approach of teaching computer programming within HEIs has a significant positive effect on students' teaching activities (Instructional Activities).	Supported
	$H_{3.4}$: The activity-based learning approach of teaching computer programming within HEIs has a significant positive effect on students' engagement (Skills Development).	Supported
	$H_{3.5}$: The activity-based learning approach of teaching computer programming within HEIs has a significant positive effect on students' feedback and assessments.	Supported
	$H_{3.6}$: Social activities has a significant positive effect on learning Gains.	Unsupported

What new framework could be used to enhance activity-based learning approach in a blended learning environment for instructing computer programming?	H_4: The developed framework has a significant positive effect on activity-based learning approach in a blended learning environment for instructing computer programming (Combined Effect).	Supported

The hypotheses derived from the research question to measure whether a construct was satisfied or not were analysed as demonstrated in Table 6.18 It was found that all the hypotheses were supported except $H_{3.6}$ i.e., Social activities have no significant positive effect on learning gains per the findings derived from the study.

6.10 A New Framework for Activity-Based Programming Instruction within a Blended Learning Environment

Finally, the research came out with a suitable framework derived from the studies for the teaching and learning of computer programming. The ABLA framework was derived from the conceptual framework of the study and data received during the three-semester period of the AR were analysed. Inputs from computer science education experts, HoDs of CS and IT departments, and the researcher's own professional experience were involved to test for all the factors that support effective teaching and learning of computer programming within a blended learning environment. The factors derived include:

1. Teaching Activities
2. Cognitive Activities
3. Social Activities
4. Activity-Based Approach Used
5. Curriculum Mediations
6. Students' Skills Development
7. Quality Assessment and Feedback
8. Learner Support
9. Technology Mediations
10. Learning Gains

Figure 6.8 denotes the Activity-Based Blended Learning Approach (ABLA) Framework for the teaching and learning of computer programming in HEIs.

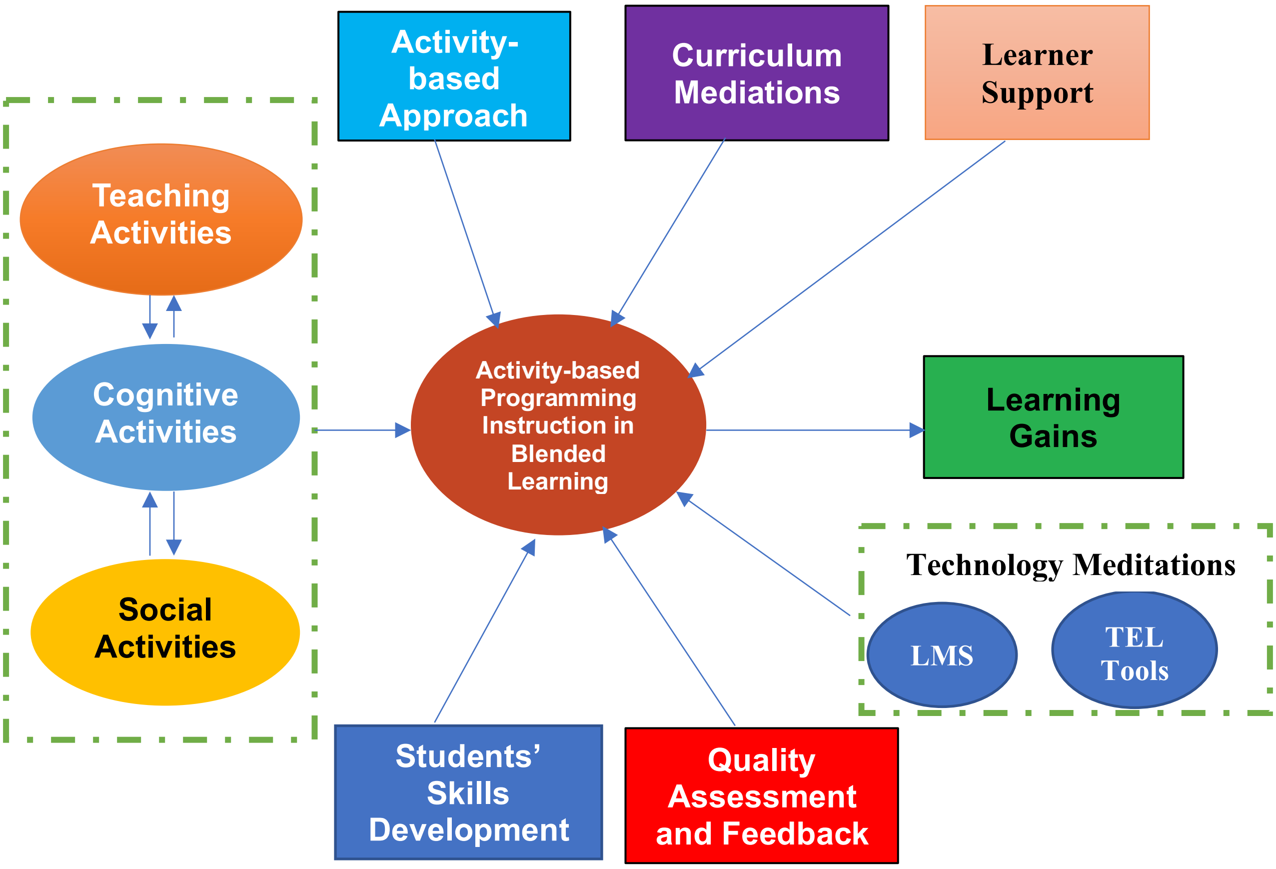


Figure 6.8: Integrated Framework for Activity-Based Programming Instruction

Following the development of the new ABLA framework after the third semester of the semester AR, the researcher reflected on the various constructs that contributes to the ABLA framework. It was noted among the experts and the researcher that the new ABLA framework has a significant positive impact on activity-based computer programming instruction within a blended learning environment and thus contributes to an effective learning gains among students. The findings contributed about 99% of the action research towards the development of the ABLA framework. The progress view of the AR for the ABLA framework is represented in Figure 6.9 below. The researcher gave room for at least 1% margin of limitations of the framework which may occur as a result of the sample size used and difference in delivery style among instructors.

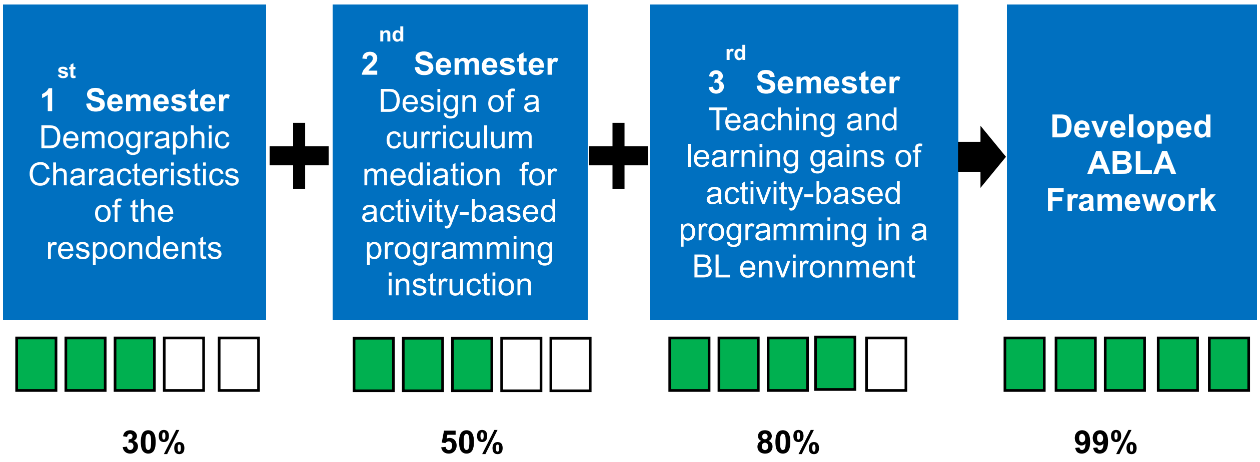


Figure 6.9 Progress View Towards ABLA Framework Development after Semester 3

6.11 Conclusion

In pursuance of the development of the activity-based framework, the researcher developed a curriculum that mediates an effective teaching and learning of computer programming within a blended learning environment. The implementation of the curriculum mediation iterated three times and was finally accepted among instructors who teach programming in the three HEIs. The revised curriculum guided the teaching and learning of computer programming and consequently, designed an activity-based programming instruction application to enhance teaching and learning. The curriculum and the ABPI system were seen to be very relevant in teaching and learning of programming in the three HEIs.

Further analysis and hypotheses test were conducted on the various effects on the key constructs i.e. instructional activities, social activities, cognitive activities, feedback on students' learning gains. The hypotheses that were derived from the research question to measure whether a construct was satisfied or not was analysed in this chapter. It was found that all the hypotheses were supported except $H_{3.6}$ i.e., Social activities have no significant positive effect on learning gains.

Findings from the interview discussions with six computer programming instructors on their experiences of the activity-based instructional approach in computer programming were ascertained and the conclusions proved very positive and relevant. The researcher finally performed a reflection on the study and consequently, developed a framework for activity-based computer programming instruction within a blended learning environment. The implications of the findings for each of the research objective per the action research is discussed in the next chapter.

CHAPTER 7

DISCUSSIONS, CONCLUSIONS AND RECOMMENDATIONS

7.1 Introduction

The general objective of this study is to develop a framework for an Activity-Based Learning Approach (ABLA) in the instruction of computer programming in a blended learning environment in higher education. In order to achieve this objective, the following specific research objectives were followed:

1. *To ascertain the current activity-based learning approaches in computer programming instruction in HEIs of Ghana within a blended learning environment.*
2. *To examine curriculum mediation based on pedagogic approaches that could support activity-based learning for instructing computer programming for HEIs in Ghana.*
3. *To investigate the learning gains of an activity-based learning approach in a blended learning environment for students and lecturers in HEIs of Ghana.*
4. *To develop a new framework that enhances an activity-based learning approach in a blended learning environment for teaching computer programming in HEIs.*

The mixed method approach was used to conduct the data analysis. During the analysis, the primary analysis, model evaluation and final analyses were done using SPSS 26 and AMOS 26 for the quantitative data and Atlas Ti for the qualitative analysis. An in-depth analysis to measure different views of the three HEIs using a one-way ANOVA test was conducted to determine how views from the three institutions differ among each other. The preliminary analysis of the study started with the evaluation of the response rate, reliability test, demographic characteristics, descriptive analysis, data screening, validation of measurement, exploratory factor analysis, confirmatory factor analysis and the analyses of hypotheses using structural equation model (SEM). The SEM was used to test for the relationships that exist between the variables and to establish the statistical model of the hypotheses.

In this chapter, a summary of the major findings of each research objectives with their discussions have been presented. The chapter also discusses the implications of the findings with respect to practice, theory and policy. Furthermore, the chapter ends with recommendations to HEIs institutions, computer science educators and the Ministry of Tertiary Education in Ghana.

7.2 Summary of Research Findings

The summary of each research findings in relation to the research objectives and research questions are discussed in this section. In an attempt to address the research gaps as underscored in the problem statement of this study, four major objectives were established for this research. The findings derived from the research objectives were set out as the foundation for deriving the implications of the study with respect to their theoretical, practical and policy implications.

7.2.1 Research Objective 1

To ascertain the current activity-based learning approaches in computer programming instruction in HEIs within a blended learning environment.

7.2.1.1 Summary of Research Findings

The research question to address the research objective 1 was:
What is the current activity-based learning approaches in computer programming instructions within a blended learning environment in HEIs?

It was also hypothesised that: H_1: The activity-based teaching and learning of computer programming in HEIs within a blended learning environment includes but is not limited to case studies, quizzes, projects, group discussions and presentations, problem-based learning and concept mapping.

A descriptive analysis, 1-Way ANOVA test and an inferential statistic using the Structural Equation Model was used to address the research questions. The findings of the qualitative data were used to equally support the quantitative data. The findings that emerged from both qualitative and quantitative analyses of the study include the following:

1. The descriptive analysis proved that the current activity-based approach used among the HEIs are Case Studies, Quizzes, Projects, Group Discussions and Presentations, Educational Games, Videos, Debates, Problem-Based Learning, Field Work, Videos, Concept Mapping and the use of TEL tools. The ANOVA test conducted shows that there are some significant differences among the approaches used among the three HEIs.
2. Most lecturers practiced activity-based computer programming instruction among the three HEIs per the adapted curriculum among the lecturers of the three HEIs.
3. The study proved that the activity-based approach used in teaching computer programming within a blended learning environment enhanced students' learning gains and experiences.

The implications of the above research findings are discussed in the next section.

7.2.1.2 Theoretical Implications of the Research

Researchers such as Chatti et al. (2010) argued that learning is individually based, socially affected, shared, found everywhere, flexible, not static and constitutes many different components. Hence, the need to diversify the delivery of teaching and learning of computer programming in Higher Education Institutions is very paramount. Consequently, the activity-based teaching and learning approach was seen to aid the diversification of delivery process with an enhanced pedagogic approach using technology as the driving force.

Again, studies from Thompson and Bell (2013) confirm that engaging students and putting students at the centre of teaching will mean that the teacher will have to discover the different needs of the students, their learning styles, their understanding of the course and motivational levels among others. In practice, this will mean that the teacher will have to devise different strategies such us the activity-based approach to meet the needs of the students with different learning styles and comprehension levels.

In view of this, the study saw a significant difference among the activity-based approaches used by the three HEIs instructors as practically understandable. This confirms the first learning principle of Merrill et al., (2003) who affirm that learning is effective when the learner is involved in the practical problem solving of a real-world situation. This principle is relevant to computer programming instruction where the learners learn how to solve real-world problems from problem analysis to implementation phases of a software.

Also, Margaryan, Collis and Cooke (2004, p. 2) asserted that the activity-based approach "*provides a way to integrate learning within students' knowledge, and, by exposing them to a variety of activities*". As a result of the high level of engagement of the learner, the study found that the activity-based approach aids both the instructor and the learner to gain a high level of interactivities thereby motivating and coaching the learners rather than just producing contents for them. The findings are also consistent with constructivist theory of Dewey (1938), Anwar (2019), Quin (2012), Kanuka and Garrison (2004); Wulf (2005) and Suhendi (2018). It was discovered from the study that the instructors were seen as the facilitators (i.e. to guide and to nuture the student) throughout the learning process and to guide the students through problem-solving procedures and activities (Anwar, 2019; Ali, 2005; Stößlein, 2009).

Generally, with regards to the students' experiences on the activity-based approach in their programming class, most students affirmed that the approach was very effective and thus, it has helped them to develop their programming skills. Also, most students responded that the activity-based programming platform is very user friendly, thus enhances their prompt responses to assignments. This supports the position of the studies of Kosterelioglu and Yapici (2016); Celik (2018); Anwar (2019); Kanuka and Garrison, (2004); Wulf, (2005) and Suhendi, (2018). Again, most students also affirmed strongly that the activity-based Programming platform made their programming experience fun; also supporting the studies around the constructivism approach in computing education (Dewey, 1938; Lister, 2007; Kosterelioglu and Yapici 2016; Asunka, Freeman and Sheeta, 2018). The findings also proved that the instructors gained an opportunity to engage with the students by assessing the academic growth of the students' learning journey through the activity-based approach which support the studies Gilboy, Heinerichs and Pazzaglia (2015); and Betihavas et al. (2015).

7.2.1.3 Practical Implications

An activity-based approach is said to occur if an instructor actively engages students in a classroom or outside the classroom to participate in the learning process through discussions, group or individual reflections, case studies, practical exercises, problem-based learning, etc.

(Anwar, 2019, Celik, 2018; Margaryan, Collis and Cooke, 2004; Hazzan, Ragonis and Lipidot, 2020). Undoubtedly, teaching and learning computer programming has for the past four decades seen a significant growth (Sharmin, 2020; Ahadi, Vihavainen and Lister, 2016) and findings from the study also confirm to this assertion. According to Alharbi and Hannaford (2011), extant literature in the STEM education reveal that Computer Science as a subject is relatively new and as such, quite limited when compared with other disciplines of Science such as Physics and Chemistry. However, the need to continue in computer science education is now more important than ever due to both societal and economic benefits it brings. A motivating factor to the computer science education especially at the undergraduate level is to produce graduates who can help in solving real-world problems (Hazzan, Ragonis and Lipidot, 2020; Bennedsen and Caspersen, 2007; Kinnunen and Malmi, 2006; Lister et al., 2004; Watson and Li, 2014; Zingaro, 2014).

In order to enhance students' learning experiences and increase their comprehension levels in computer programming constructs, there was a need for the researcher to ascertain the problems associated with the reasons why students are unable to code or perform well in computer programming courses as lucidly mentioned in the studies of Ahadi, Vihavainen and Lister (2016); Lister et al. (2007) Sarpong, Arthur and Amoako (2013)

It was lucidly noted that most students are really suffering from understanding programming language syntax and constructs per the findings derived from the study. This is an affirmation of the already defined challenges that most students asserted in the quantitative and qualitative data. Some students asserted that understanding the practical sessions of programming were challenging and some students also indicated that programming concepts have always been difficult to understand. Some students reported that lecturers minimally issued feedbacks from their programming assignments and exercises which also affected their comprehension levels. The situation is not contrary to the design of a program to solve a real-world problem, dividing functionalities to solve procedures, and finding bugs and debugging their own codes. Thus, most of the students responded in the affirmative on these prevailing challenges as indicated section 5.2.8.

Interestingly, the findings from the three HEIs were not different from each other with respect to the preponderant challenges the students faced in the programming classes as discovered from the one-way ANOVA test and the SEM model. It was noted that there was statistically significant difference between the three HEIs, since the p-value was greater than 0.05. The p-value was 0.49 among the three institutions. Again, this research supports studies of Kinnunen and Malmi (2006); Lister et al., (2004); Watson and Li, (2014); Zingaro (2014). Overmars (2004); Ali, (2005); Linn (2009) ; Sarpong, Arthur and Amoako, 2013). The overall impression from the study indicates that there is little application on the activity-based approaches used in the teaching and learning of computer programming in HEIs following the above worrying trend.

However, findings from the six instructors and some students attributed the relevance of the activity-based learning approach as an effective pedagogic method within a blended learning environment that supported their teaching and learning experiences. It was noted from the findings that activity-based approaches such as group activities, problem-solving, case studies, video, games, the use of TEL tools, and more contact support with tutors and lecturers with students in class and after class was seen as an effective practice that fostered students' programming experience. These findings have got practical implications of the study and resonates with the previous studies of Dewey (1938); Piaget (1950); Bruner (1996); Ben-Ari (1998); Grover and Pea (2013); Walker, Voce and Ahmed (2012); Lister et al. (2004); and Watson and Li (2014).

Another emerging discovery from the qualitative analysis was the provisioning of learner support as posited by Ahsan and Mullick, (2013), Usum, (2012) and Chattopadhyay (2014). It was discovered that students lacked support after their classroom activities. Thus, students attributed their poor programming skills to the truncated engagements beyond the classroom environment. Also, some students reported that most of their programming lecturers were not competent enough to teach. This alarming feedback drew the attention of the researchers to probe the students further. The students further indicated that their lecturers only came to the class to teach theories and do not engage them to practically solve problems. Hence, the findings from this study have equally got practical implications on both the students and the lecturers to provide learner support irrespective of the activity-based approach used for teaching and learning.

Again, it was deduced from the findings that the use of videos and Technology Enhanced Learning (TEL) tools such as, Socrative, Padlets and Google Docs, was noted to be useful tools that enhance the comprehension of programming. In the studies of Walker, Voce and Ahmed (2012), they posit that using a TEL approach in learning supports active engagement by students and also enhances their performance and learning experience. Moreover, the use of educational games, programming debates, round table discussions of critical issues, peer review, field work and concept mapping were rarely practiced in the teaching and learning of computer programming among the three institutions. It was, however, discovered that instructors who employed activity-based approaches such as educational games, programming debates, round table discussions of critical issues, peer review, field work and concept mapping saw a practical significance in teaching and learning of programming.

7.2.1.4 Policy Implications

The findings of the research objective 1 show the need for HEIs academic administrators, Computer Science Faculty/Divisions/Departments, the Ministry of Education in Charge of Tertiary, the National Council for Tertiary Education and The National Accreditation Board to make reforms in the pedagogic approaches for teaching and learning of computer programming (Lister et al., 2004; Watson and Li, 2014; Zingaro, 2014).

The analyses of the qualitative data from both the lecturers and students shed light on some of the factors identified as contributing to the poor performance of students in computer science education which literature also supports. These factors include; lecture attendance and contact sessions to students, and poor team work among students (Sheard, Carbone, and Laakso, 2010), work overload, that is, too much credit hours assigned to students and lecturers (Rountree et al., 2004); students' self-efficacy and confidence in the ability to program (Lewis and Loftus, 2009) and perceived value to computer science to solving real world issues (Biggers, Brauer, and Yilmaz, 2008). The findings from the study resonate with the previous studies as underscored and thus requires a policy reforms to unravel this research gap.

In a different study, Lang et al. (2007) discovered that factors that influence the rate of retention in computer science education include gender factors in which females are more affected and then pedagogical factors which are associated with the curriculum and assessment. Petersen et al. (2016) and Kinnunen and Malmi (2006) opined that lack of time and motivation, lack of learner support, poor time management skills, perceived difficulty of the computer science course and lack of interest in the computer science education among students contribute to the poor performance which needs an extensive policy consideration and reformation.

7.2.2 Research Objective 2

To examine curriculum mediation based on pedagogic approaches that could support activity-based learning for instructing computer programming for HEIs in Ghana.

7.2.2.1 Summary of Research Findings

The research question to address the research objective 2 was:
What curriculum mediation based on pedagogic approaches could support activity-based learning for instructing computer programming in HEIs of Ghana?

It was also hypothesised that: H_2: The developed curriculum used in teaching computer programming within a blended learning environment supports activity-based learning.

A descriptive analysis, 1-Way ANOVA test, Rapid Application Development (RAD) approach, and qualitative analysis were used to address the research questions. The findings of the qualitative data using content and thematic analysis were used to equally support the quantitative data.

The findings that emerged from both qualitative and quantitative analyses of the study include the following:

1. The developed curriculum that mediates an effective teaching and learning of computer programming within a blended learning environment was very relevant.
2. The developed activity-based Programming Instructional (ABPI) system was discovered to be a user-friendly system that supports blended learning approach. The ABPI system was found to be very relevant in the teaching and learning of programming and thus, support the comprehension of computer programming among students.

3. The curriculum developed supported the activity-based instructional approach within a blended learning environment and thus, aid in the quality assessment and feedback practices.

The implications of the above research findings are discussed in the next section for the research objective 2.

7.2.2.2 Practical Implications on the Findings

On the practical implications on the curriculum mediations, it was found that the curriculum/syllabus/course developed for the teaching and learning of computer programming was relevant for the instructors and supported the activity-based approach in the instructors' institutions as posited by Celik (2018); Quin (2012); Kosterelioglu and Yapici (2016); Kinnunen and Malmi (2008). The pedagogic approach of using the activity-based approach was highly recommended by five instructors and confirmed that it should be encouraged in all computing science courses delivery within a blended learning environment which resonates in the study of Shah and Rahat (2014); Lang et al. (2007); Huang, Ma, and Zhang (2008); Shah and Rahat (2014) and Singal et al. (2018).

The assessment methods for teaching and learning of programming was also keenly considered as posited by Boud and Associates (2010); Carless et al., (2011). Findings from the study affirmed that formative assessment method as Price and Kirkwood (2011) explained, should carry greater assessment percentage (e.g. 70% or 60%) of the total course assessment while summative assessment carries the remaining 30% or 40% to foster the activity-based approach. The findings on the formative assessment method conforms to best practices, standards and good assessment criteria (Boud and Associates, 2010; Carless et al., 2011; Fluckiger et al., 2010; Gibbs and Simpson, 2004; Gilbert et al., 2011; Miller et al., 2010; Sadler, 2010).

7.2.2.3 Policy Implications of the Findings

In the study of Price and Kirkwood (2011), it was discovered that assessment and feedback work together to effectively enhance quality delivery of assessment to students. The 'who' and 'why' for the quality assessment was discovered to influence policy in the teaching and learning of computer programming within a blended learning environment. In view of this, the study sought the approval of the Faculty board to test the reliance of the curriculum developed and was used during and after the delivery of the course. It was discovered that Technology Enhanced Learning (TEL) tools that support quality assessment and feedback practices included; for example, Turnitin, Moodle, Socrative, Screen Cast O-matic and Sakai.

Mohanty and Vohra (2006); Price and Kirkwood (2011) posit that higher education institutions must employ the use of Turnitin to check plagiarism or similarity index of students' assessment submissions. Hence, HEIs managers and administrators, the National Accreditation Board, the Ministry of Education and the National Council for Tertiary Education must see to it that there is a high standard of assessment practices among HEIs in Ghana. For instance, Turnitin and Moodle were seen to support the provision of assessments and feedback to students and also aid in both summative and formative assessments (Price and Kirkwood, 2011). Lecturers can further give instant feedback to the students using text, audio or recorded video to grant feedback to the student(s) via the LMS and the developed ABPI. Also, examples of assessment tools that supported audio-visual feedbacks are Adobe Spark, Screen Cast O-matic and Padlet.

7.2.3 Research Objective 3

To investigate the learning gains of an activity-based learning approach in a blended learning environment for students and lecturers in HEIs of Ghana.

7.2.3.1 Summary of Research Findings

The research question to address the research objective 3 was:
What are the teaching and learning gains of an activity-based learning approach in a blended learning environment for students and lecturers in HEIs? To answer the research objective 3, the researcher took consideration in the various factors that could affect learning gains following the conceptual design of the framework.

It was also hypothesised that:

1. H_3: The activity-based learning approach of teaching programming within HEIs has a significant positive effect on students' learning gains.

2. $H_{3.1}$: The activity-based learning approach of teaching programming within HEIs has a significant positive effect on students' cognitive development.
3. $H_{3.2}$: The activity-based learning approach of teaching programming within HEIs has a significant positive effect on students' social activities and learning.
4. $H_{3.3}$: The activity-based learning approach of teaching programming within HEIs has a significant positive effect on students' teaching activities (Instructional Activities).
5. $H_{3.4}$: The activity-based learning approach of teaching programming within HEIs has a significant positive effect on students' engagement (Skills Development).
6. $H_{3.5}$: The activity-based learning approach of teaching programming within HEIs has a significant positive effect on students' feedback and assessments.
7. $H_{3.6}$: Social activities has a significant positive effect on learning gains.

A descriptive analysis, 1-Way ANOVA test, Structural Equation Model (SEM), and qualitative analysis were used to address the research questions. The findings of the qualitative data were conducted using content and thematic analyses to equally support the quantitative data.

The findings that emerged from both qualitative and quantitative analyses of the study include the following:

1. It was shown in the study that the activity-based learning approach of teaching programming within HEIs had a significant positive effect on the learning gains ($r=0.835$, $p<.000$). A unit increase in activity-based learning approach of teaching programming within HEIs results in 85% increase in the student learning outcomes.

2. The study also found that the activity-based learning approach of teaching programming within HEIs has a significant positive effect on students' cognitive development. This finding was supported with effect of $r=0.880$, $p<0.000$. This is to say a unit change in the activity-based learning approach of teaching programming within HEIs results in an 88% change in the student's cognitive development.

3. The study found that the activity-based learning approach of teaching programming within HEIs has a significant positive effect ($r=0.7847$, $p<0.000$) on students' social activities and learning. This was found to be significant. Again, this to say a unit

increase in the activity-based approach of teaching programming will result in a 78.5% increase in the student's social activities.

4. Findings from the study also affirms that the activity-based learning approach of teaching programming within HEIs has a significant positive effect on instructional activities with ($r=0.850$, $p<0.000$). This implies that a unit change in the activity-based learning approach of teaching programming within HEIs will result in an 85% increase in instructional activities.

5. The activity-based learning approach of teaching programming within HEIs has a significant positive effect on the skills development of students. This was also found to be significant: ($r=0.871$, $p<0.000$). Again, unit change in the activity-based learning approach of teaching programming within HEIs will hence, result in an 87.1% increase in skills development.

6. It was also found that the activity-based learning approach of teaching programming within HEIs has a significant positive effect on students' feedback and assessments, significant with ($r=0.866$, $p<0.000$). This means that a unit increase in the activity-based learning approach of teaching programming within HEIs will result in an 86.6% increase in student's feedback and assessment.

7. Finally, it was found that social activities have no significant positive effect on students' learning gains ($r=-0.083$, $p=0.274$). This means that a unit increase in social activities of teaching programming within HEIs will result in an 8.3% increase in student's learning gains.

The implications of the above research findings regarding research objective 3, are discussed in the next section.

7.2.3.2 Practical Implications on the Findings

Millheim, (2012) asserted that an effective teaching and learning is for the student to gain a reasonable knowledge that can be applied in a useful manner to solve real world problems. It was found that the activity-based teaching and learning of computer programming within a blended learning environment had positive effect on the students' academic performance as well as gaining knowledge in their skills for the job market; thus, contributing to the practical implications of the study. This affirms the studies of Anwar (2019); Chew, Jones and Turner (2008); Freitas and Leonard (2011); Millheim (2012). Also, the LMS platform used was seen to a critical tool for teaching and learning in HEIs (Matarirano et.al, 2021).

Again, it was discovered that the activity-based approach of teaching and learning of programming was the rationale for a successfully academic growth of the students' learning journey which confirmed the study of; Gilboy, Heinerichs, and Pazzaglia (2015); Betihavas et al., (2015). Also, it was discovered that instructional activities and cognitive activities affected the comprehension levels of computer programming and consequently enhanced the learning gains of students; thus, the study provides a practical implication for practice and adoption. This supports the studies of Garrison and Vaughan (2008); Garrison, Anderson and Archer (2010); Szeto (2015b); Chew, Jones and Turner (2008); Freitas and Leonard (2011); Millheim, (2012).

7.2.3.3 Theoretical Implications

The activity-based instructional approach for teaching computer programming within a blended learning environment was seen as a relevant pedagogic approach in Computer Science education. In the light of this, the researcher employed the community of enquiry (see p.113) model as posited by Garrison, Anderson and Archer (2010); Szeto, (2015); Chew, Jones and Turner (2008). It was discovered from the findings that the instructional and cognitive activities, activities, feedback and assessment, and students' skills development affected positively the students' learning experiences of computer programming. These findings supported the assertions by several researchers such as, Chew et al., (2008); Anderson et al., (2001); Cavalcante, Riberas and Rosa, (2016) who also agreed that the Community of Inquiry (COI) model by Garrison, Anderson and Archer (2010) is a very effective model for implementing blended learning in higher education course delivery. This, contributes to the theory that supports teaching and learning of computer programming within a blended learning environment.

It was also discovered in the structural equation model that social activities that were used in the COI model did not have significant effect on the students' learning gains which also did not support the hypothesis. However, it was seen to support other factors such as cognitive and skills development for activity-based programming delivery within blended learning environment. This also supports Garrison, Cleveland-Innes and Fung (2010); and Szeto, (2015a) on the social activities' indications of the COI framework affirming that there is a strong relationship between teaching activities and cognitive activities. In other words, the social activities served as a mediating paradigm existing between the teaching activities and cognitive activities. This is because teaching and learning are seen as social activities.

7.2.3.4 Policy Implications

The study has shed light on the need to make reforms in the delivery of computer programming by using the activity-based pedagogic approach to avert the high failure rates, poor performance and lack of programming skills among undergraduate students in HEIs. Hence, HEIs academic administrators and Computer Science Faculty/Divisions/Departments, the Ministry of Education in Charge of Tertiary Education, the National Council for Tertiary Education and The National Accreditation Board, to make reforms in the pedagogic approaches for teaching and learning of computer programming (Lister et al., 2004; Watson and Li, 2014; Zingaro, 2014).

7.2.4 Research Objective 4

To develop a new framework that enhances an activity-based learning approach in a blended learning environment for teaching computer programming in HEIs.

7.2.4.1 Summary of Research Findings

The research question to address the research objective 4 was:

1. What new framework could be used to enhance activity-based learning approach in a blended learning environment for instructing computer programming?

It was also hypothesised that H4:

The new developed framework has a significant positive effect on activity-based learning approach in a blended learning environment for instructing computer programming.

A descriptive analysis, 1-Way ANOVA test, the Structural Equation Model, and qualitative analysis were used to address the research questions. The findings of the qualitative data using content and thematic analyses were used to equally support the quantitative data to develop the framework. It was found that:

1. There is a significant positive effect of the framework of the four activities (Instructional activities, Social activities, Cognitive activities, Activity-based learning used) on learning goals with a standardized coefficient, = 0.835, $p<0.01$.
2. It was also noted that the activity-based approach in teaching computer programming with a blended learning environment (i.e. all combined activities) have a significant impact on learning, except the social activities.
3. Lastly, it was found out that social activities alone did not have a significant impact on gains made in learning. In other words, being socially active in class does not necessary guarantee learning gain.
4. The developed curriculum that mediates an effective teaching and learning of computer programming within a blended learning environment had significant effect on student's learning experiences.
5. The curriculum developed supported the activity-based instructional approach within a blended learning environment and thus, aided in quality assessment and feedback practices.

The developed framework has implications for the theoretical, policy and practical effects for the teaching and learning of computer programming with a blended learning environment. The discussions of the implications of the study are seen in the previous practical, policy and theoretical implications for research objectives 1, 2 and 3 respectively.

7.3 Literature Gaps Addressed

Extant literature such as Nouri, (2016) and Voronina et al., (2017), strongly criticized teaching and learning inefficiencies in HEIs. It was established in literature that most HEIs are still using the traditional lecture halls, lecture materials and conventional assessment methods (i.e. summative assessments). Findings from the study affirmed that the blended learning approach avert issues associated with the limitations associated with the traditional classroom environment with larger population of students with minimal students' engagements in Ghana. This research findings supports previous studies from Nel (2017); Kintu, Zhu and Kagambe (2017), Huang, Ma and Zhang (2008), Voronina et al., (2017).

Also, the studies of Greyling et al. (2006), Saeli et al. (2011), Ala-Mutka, (2003); Overmars, (2003); Ali (2005); Linn (2009); Sarpong, Arthur and Amoako, (2013); Lister et al., (2005); Watson and Li (2014); and Zingaro (2014), congruently asserted some identifiable research gaps that most teachers face when teaching computer programming language in HEIs. Among these research gaps are included issues in understanding computer programming syntax and logics correctly, problem-solving capabilities in the software development, and poor performance in computer programming courses among students.

Findings from the study indicated that factors contributing to avert the poor performance, lack of skills to code and students' inability to comprehend computer programming syntaxes and semantics as denoted in previous literature include the:

- Adoption of activity-based computer programming framework that supports teaching and learning in HEIs within a blended learning environment (Garrison, Anderson and Archer, 2010; Lister et al., 2005; Watson and Li, 2014; Zingaro, 2014).
- Adoption of activity-based pedagogic approaches in teaching and learning of computer programming among undergraduate students within a blended learning environment (Harris, Mishra and Koehler, 2008; Oliva and Gordon, 2012; Pinar, 2012).
- Design of a curriculum mediation to support the delivery of computer programming within a blended learning environment (Price and Kirkwood, 2011; Robins, Rountree and Rountree, 2004; Kinnunen and Malmi, 2008; Boud and Associates, 2010; Carless et al., 2011; Garrison and Vaughan, 2008; Szeto, 2015).

Finally, researches by Greyling et al. (2006), Cavalcante, Riberas and Rosa (2016); Nel (2017); Ahsan and Mullick (2013); Anderson and Elloumi (2004); and Dietinger (2003) asserted that one critical factor resulting in teaching and learning computer programming is the provisioning of learner support and frequent students' engagement. It was, however, established from the studies that, the development of Activity-Based Programming Instructional (ABPI) platform provides a solution to the research gaps with a user-centred platform that interfaces the engagements between instructors and students.

7.4 Summary of Research Objectives, Methodology, Findings, Implications and Research Gaps Addressed

In this section, the summary of the research findings, the research gaps and the summary of the implications of the study have been outlined Table 7.1 below.

Table 7.1: Summary of Research Objectives, Methodology, Findings, Implications and Research Gaps Addressed

No.	Research Objectives	Types of Analytical Techniques used	Summary of Findings	Implications of Findings	Research Gap Addressed
1	To ascertain the current activity-based learning approaches in computer programming instruction in HEIs within a blended learning environment.	Descriptive Analysis, 1-Way ANOVA Test, Multiple Regression Analysis, Thematic Analysis, Pattern Matching, and Logical Narrations.	The study ascertained that activity-based teaching and learning of computer programming in HEIs within a blended learning environment include but not limited to case studies, quizzes, projects, group discussions and presentations, problem-based learning, games the use of TEL tools, videos and concept mapping. Also, the study proved that the activity-based approach used in	1. Practical implications 2. Policy implications 3. Theoretical implications	The study has averted issues in literature as indicated in the studies of Greyling et al. (2006), Saeli et al. (2011), Ala-Mutka (2003); Overmars (2003); Ali (2005); Linn (2009); Sarpong, Arthur and Amoako, 2013); Lister et al., (2005); Watson and Li (2014); and Zingaro (2014) on the adoption and practice of activity-based teaching and learning approach.

			teaching computer programming within a blended learning environment enhanced students' gains in learning and experiences.		
2	To develop a curriculum mediation based on pedagogic approaches and application that could support activity-based learning for instructing computer programming for HEIs in Ghana.	Action Research Approach (Teaching, Observation and Reflection) Rapid Application Development (RAD) Thematic Analysis, Pattern Matching, and Logical Narrations.	The new curriculum developed was found to support effective teaching and learning of programming using activity-based approach within the blended learning environment.	1. Practical implication 2. Policy implication	The study also shed light on the need to improve upon the curriculum used in the teaching and learning of computer programming. Hence, averting or resolving issues in literature as indicated in the studies of Szeto (2015); Price and Kirkwood (2011); Nel (2017); Robins, Rountree and Rountree, (2004); Kinnunen and Malmi (2008); Boud and Associates (2010); Carless et al. (2011); and Garrison and Vaughan (2008).

3	To investigate the learning gains of an activity-based learning approach in a blended learning environment for students and lecturers in HEIs of Ghana.	Descriptive Analysis, 1-Way ANOVA Test, Multiple Regression Analysis, Structural Equation Model, Exploratory Factor Analysis, Confirmatory Factor Analysis, Thematic Analysis, Pattern Matching, and Logical Narrations.	1. The study showed that activity-based learning approach of teaching programming within HEIs in Ghana had a significant positive effect on the learning gains. 2. The study also found out that activity-based learning approach of teaching programming within HEIs in Ghana has a significant positive effect on students' cognitive development. 3. The study found that activity-based learning approach of teaching programming within HEIs in Ghana has a significant positive effect on students'	1. Practical implications 2. Theoretical implication 3. Policy implications	The findings from the studies have also provided solutions that obviates various issues associated with the comprehension levels, learning gains and skills development of computer programming among students. It was however noted that the application of the COI model (Garrison, Anderson and Archer, 2010) in its entirety was not seen to fully support the activity-based teaching and learning of computer programming. Aspects such as learning gains, quality assessment and feedback, and skills development were seen as research gaps that the findings

			social activities and learning. 4. The activity-based learning approach of teaching programming within HEIs in Ghana has a significant positive effect on instructional activities. 5. Activity-based learning approach of teaching programming within HEIs in Ghana has a significant positive effect on the development of skills of students. 6. Activity-based learning approach of teaching programming within HEIs in Gnana has a significant positive effect on students' feedback and assessments		from the studies provided solution to. Hence, findings from the studies finally averted literature gaps in the studies of Millheim (2012), Garrison, Anderson and Archer (2010). Harris, Mishra, and Koehler (2008); Oliva and Gordon (2012); Pinar, (2012), McLeod (2007); Chew, Jones and Turner (2008); and Freitas and Leonard (2011).

			7. It found that social activities have no significant positive effect on students' learning gains.		
4	To develop a new framework that enhances an activity-based learning approach in a blended learning environment for teaching computer programming in HEIs.	Structural Equation Model, Thematic Analysis, Pattern Matching, and Logical Narrations.	The developed framework has significant positive effect on activity-based learning approach in a blended learning environment for instructing computer programming.	1. Practical implications 2. Policy implication 3. Theoretical implication	The findings from the studies have shed light on the need to adopt a framework for the teaching and learning of computer programming in HEIs within a blended learning environment. The developed framework consequently, averts research gaps in the teaching and learning of computer programming as seen in the studies of Hazzan, Ragonis and Lipidot (2020), Ali (2005), Byrne, Fisher, and Tangney (2016), Chew, Jones and Turner (2008a); Li (2010); Overmars (2004)

7.5 Recommendations

From the discussions and conclusions arrived at, the researcher proffers some recommendations.

It is recommended:

1. That the activity-based instructional approach for teaching computer programming within a blended learning environment should be used in all the teaching and learning processes of computing science education. Again, a reformation in the computer programming curricula among HEIs to reduce the high failure rates, poor performance and lack of programming skills among undergraduate students in HEIs needs to be reformed.
2. That HEIs academic administrators and Computer Science Faculty/Divisions/Departments, the Ministry of Education in Charge of Tertiary Education, the National Council for Tertiary Education and The National Accreditation Board, make reforms in the pedagogic approaches for teaching and learning of computer programming to enhance students learning gains, programming skills and employability opportunities among undergraduate students.
3. That HEIs academic administrators and Computer Science Faculty/Divisions/Departments increase the contact sessions between the students and the instructors.
4. That HEI Management should improve the provision of constant learner support and prompt feedback to students.
5. That Higher Education Managers, Administrators, the National Accreditation Board, the Ministry of Education and the National Council for Tertiary Education set up high standards of assessment at all levels of computing science education programmes among HEIs in Ghana.
6. That the developed and tested framework for the activity-based programming instruction within a blended learning environment be adopted and practised among all higher education institutions in Ghana.

7.6 Suggestions for Future Research

As associated with scientific studies, the research provides significant areas for further studies. It is suggested that further research be conducted in the influence of activity-based computer programming learning approach on students' employability and socio-economic benefits. Again, it is suggested that a country-wide study on the impact of activity-based instructional approach in computer programming instruction on students learning performance and experience should be undertaken.

Further, this study did not involve HEIs administrators in their institutions' adoption and practice of the activity-based instructional approach of teaching and learning of computer programming within a blended learning environment among undergraduate students. Further studies in that area can be conducted.

7.7 Conclusion

This study has presented the various literature, methodology, major findings, discussions, summary of major findings of this study with respect to the research objectives and questions and recommendations to key stakeholders in HEIs in Ghana. The implications of the findings of the research objectives in relations to theory, policy and practice have been discussed. The study developed a framework for an activity-based computer programming instructional approach in a blended learning environment of higher education institutions in Ghana. The motivating factor to conduct this study is the poor performance of most undergraduate students in the domain of computing science programs to comprehend and write computer programs to solve real-world problems. In addition, a fundamental problem with most HEIs in undergraduate computer science educational programmes is how to teach programming and how students can comprehend and write software programs.

The methodology used to conduct the research is the action research approach which spanned through three different semesters from September 2019 to October 2020. The mixed method approach was used to capture data from 300 students and nine lecturers in three different HEIs in Ghana and different computer programming classes within a blended learning environment.

Ultimately, the findings from the study provided a framework indicating the various components of activity-based computer programming instructional approach within a blended learning environment that enhances learning gains for students. Firstly, the demographic information gathered informed the researcher about the characteristics of the respondents, their perception of active-based teaching and learning and computer programming instruction during the first semester of the AR. Afterwards, a curriculum mediation was developed to guide the teaching and learning of computer programming using the activity-based approach within a BL environment during the second semester. In each of the semesters, different instruments were used to collect the data towards the development of the ABLA framework. In the third semester, it was found that instructional activities, cognitive activities, skills development, assessment and feedback, curriculum and technology mediation, and learner support activities have positive significant effect on students' learning gains.

The results from both the quantitative and qualitative aspect of the data analysis found that the activity-based teaching and learning of computer programming within a blended learning environment aids to avert the challenges among undergraduate students' comprehension of computer programming syntax and semantics. Moreover, the study found that students' learning journey of computer programming comprehension needs continuous learner support and engagements using activity-based instructional approach within a blended environment.

Finally, both the quantitative and qualitative aspect of the data analysis found that teaching and learning programming was seen as a major challenge among undergraduate students (i.e. the understanding of programming syntax and semantics). Also, a critical aspect of the study also found that student's learning journey of computer programming comprehension requires continuous learner support and engagements using the activity-based instructional approach within a blended environment. Therefore, recommendations to HEIs managers, administrators, the National Accreditation Board, the Ministry of Education and the National Council for Tertiary Education need to make provisioning for strategic policies to reform the teaching and learning of computer programming to develop students that can transform the world through technology and software solutions.

References

Abbotts, R., Bolton J., Brown, K., Fox, J., Hawkes, L., Jones, C., Joyce, L., McGimpsey, H., Sharman R., Stepney, M., Thacker S., Tunnicliffe, P., and Watts, S. (2008) *Making Waves in Education*. York: The University of York. URL: http://dera.ioe.ac.uk/id/eprint/13075.

Adams, P., (2006) Exploring social constructivism: theories and practicalities, *Education,* 3-13, 34:3, 243-257, DOI: 10.1080/03004270600898893.

Adnan, M. and Anwar, K. (2020) Online Learning Amid the COVID-19 Pandemic: Students' perspective. *Journal of Pedagogical Sociology and Psychology*, 2(1), pp. 45–51, URL: http://www.doi.org/10.33902/JPSP. 2020261309.

Ahadi, A., Vihavainen, A. and Lister, R. (2016) On the number of attempts students made on some online programming exercises during semester and their subsequent performance on final exam questions. *ACM Conference on Innovation and Technology in Computer Science Education*, pp. 218–223.

Ahsan, M. T. and Mullick, J. (2013) The journey towards inclusive education in Bangladesh: Lessons learned, *Prospects*, 43(2), pp. 151–164. Available at doi: 10.1007/s11125-013-9270-1.

Akyol, Z. and Garrison, D. R. (2011) Understanding cognitive presence in an online and blended community of inquiry: Assessing outcomes and processes for deep approaches to learning, *British Journal of Educational Technology*, 42(2), pp. 233–250. Available at doi: 10.1111/j.1467-8535.2009.01029. x.

Al-Imamy, S., Alizadeh, J. (2006) On the Development of a Programming Teaching Tool: The Effect of Teaching by Templates on the Learning Process, *Journal of Information Technology Education*, 5, pp.271-283.

Ala-Mutka, K. (2004) Problems in learning and teaching programming - a literature study for developing visualizations, Codewitz-Minerva project. Available at https://www.cs.tut.fi/~edge/literature_study.pdf. Accessed on September 23, 2019.

Alammary, A. (2019) Blended learning models for introductory programming courses: A systematic review. *PLoS ONE* 14(9).

Alharbi, A., Henskens, P. F and M. Hannaford, M. (2011) Pedagogical Framework to Improve the Quality of Learning Objects in Collaborative e-Learning Systems, *3rd International Conference on Computer Supported Education*, Noordwijkerhout, Netherlands.

Alharbi, A. D. Henskens, P. F. and Hannaford, M. (2011) An Investigation into the Learning Styles and Self-Regulated Learning Strategies for Computer Science Students. *Proceedings of Ascilite 2011,* University of Tasmania, Hobart.

Alhazmi, S., Hamilton, M., and Thevathayan, C. (2018) Cs for all: Catering to diversity of master's students through assignment choices. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education - SIGCSE 18*, 38–43. doi: 10.1145/3159450.3159464.

Ali, S. (2005) Effective Teaching Pedagogies for Undergraduate Computer Science, *Mathematics and Computer Education*, 39(3), pp. 243–257.

Alkhatib, O. J. (2018) An interactive and blended learning model for engineering education, *Journal of Computers in Education*. Springer Berlin Heidelberg, 5(1), pp. 19–48. doi: 10.1007/s40692-018-0097-x.

Alturki R.A. (2016) Measuring and Improving Student Performance in An Introductory Programming Course. *Informatics in Education.* 15(2), pp.183–204.

Alzaghoul, A. F. (2012) The Implication of the Learning Theories on Implementing E-Learning Courses, *The Research Bulletin of Jordan ACM*, Vol. II, pp. 27–30.

An, Y.-J. (2013) Systematic Design of Blended PBL: Exploring the Design Experiences and Support Needs of PBL Novices in an Online Environment, *Contemporary Issues in Technology and Teacher Education*, 13(1), pp. 61–79. Available at: http://www.editlib.org/p/42059.

Anderson, T and Elloumi, F. (2004) *Theory and Practice of Online Learning*, Athabasca University, Athabasca, AB T9S 3A3-Canada.

Anderson, T., Rourke, L., Garrison, D. (2001) Assessing teaching presence in a computer conferencing context, *Journal of Asynchronous Learning Network*, 5(2), pp. 1-17. Available at: doi: 10.1.1.95.9117.

Anwar, F. (2019) The effect of activity-based teaching techniques on student motivation and academic achievement. *Journal of Education and Educational Development*, 6(1), pp. 154-170

Arbuckle, J.L., (2013) *Amos 22 user's guide*. Chicago, Illinois: SPSS.

Asunka, S., Freeman, E., Sheeta, L. A. (2018) Implementing Constructivist Pedagogy in a Flipped Mode in a Postgraduate Course. *Proceedings of ICERI2018 Conference,* 12th-14th November 2018, Seville, Spain.

Babbie, E. (2004) *The Practice of Social Research*. 10th edn. California: Thomson Wadsworth.

Barrows, H., S. (1996). Problem-based learning in medicine and beyond: A brief overview. *New Directions for Teaching and Learning*. 1996 (68), pp. 3-12. doi:10.1002/tl.37219966804.

Baruch, Y. and Holtom, B. C. (2008) Survey response rate levels and trends in organizational research. *Human Relations*, 61(8), pp. 1139 - 1160. https://doi.org/10.1177/0018726708094863.

Baskerville, R. and Pries-Heje, J. (1999) Grounded action research: a method for understanding IT in practice. *Accounting, Management and Information Technology*, 9(1), pp. 1-24.

Baskerville, R.L. (1999) Investigating information systems with action research, *Communications of the Association for Information Systems*, 2(19) Article 19. DOI: 10.17705/1CAIS.00219.

Baty, P. (2006) *Sausage factory culture puts the squeeze on ideas*, London: Times Higher Education Supplement. Retrieved 11 Sept 2008, Available from: http://www.timeshighereducation.co.uk/story.asp?storyCode=202912&sectioncode=26.

BBC (2004) *Problems at e-learning University*, Retrieved 10 September 2019, Available from: http://news.bbc.co.uk/1/hi/education/3494048.stm.

BBC (2006) *Oppression of the Public Universities in Malaysia*, Retrieved 11 April 2019.

Ben-Ari, M. (1998) Constructivism in Computer Science Education. *ACM SIGCSE Bulletin*, 30(1), pp. 257–261. https://doi.org/10.1145/274790.274308.

Bell, T., Alexander, J., Freeman, I., and Grimley, M. (2009) Computer Science Unplugged: School Students Doing Real Computing Without Computers. *New Zealand Journal of Applied Computing and Information Technology*, 13(1), pp. 20-29.

Bennedsen, J. M. and Caspersen, M.E., (2007) Failure Rates in Introductory Programming. *ACM SIGCSE Bulletin*. 32-36, https://doi.org/10.1145/1272848.1272879.

Bentler, P.M. (2005) *EQS 6 Structural Equations Program Manual.* Encino, CA: Multivariate Software, Inc. ISBN 1-885898-03-7.

Bereiter, C. (1994) Constructivism, Socio-culturalism, and Popper's, *World Educational Researcher*, 23(7), pp. 21-24.

Betihavas, V., Bridgman, H., Kornhaber, R., and Cross, M. (2015) The evidence for flipping out: A systematic review of the flipped classroom in nursing education. *Nurse Education Today*, 6, pp. 15–21.

Bigge, M. L. and Shermis, S.S. (1992) *Learning theories for teachers*. 5th edn. New York: Harper Collins Publishers.

Biggers, M., Martinelli, A. and Yilmaz, T. (2008) Student perceptions of computer science: a retention study comparing graduating seniors with CS leavers, *Proceedings of the 39th SIGCSE*

Technical Symposium on Computer Science Education, SIGCSE 2008, DOI: 10.1145/1352135.1352274.

Blanco, Q. A., Carlota, M. L., Nasibog, A. J., Rodriguez, B., Saldaña, X. V., Vasquez, E. C., and Gagani, F. (2020) Probing on the Relationship between Students' Self-Confidence and Self-Efficacy while engaging in Online Learning amidst COVID-19, *Journal La Edusci*, 1(4), pp. 16–25. doi: 10.37899/journallaedusci.v1i4.220.

Blank, G. D., Roy, S., Sahasrabudhe, S., Pottenger, W. M., and Kessler, G. D. (2003) Adapting multimedia for diverse student learning styles. *Journal of Computing Sciences in Colleges*, 18(3), pp. 45–58.

Bosch Software Inc., (2019) *How to approach smart learning*, Available at: https://blog.bosch-si.com/future-of-work/how-to-approach-smart-learning-environments/ Accessed: 5th August, 2019.

Brahimi, T. and Sarirete, A. (2015) Learning outside the classroom through MOOCs, *Computers in Human Behavior.* Elsevier Ltd, 51, pp. 604–609. doi: 10.1016/j.chb.2015.04.014.

Bryman, A. and Bell, E., (2015) Business Research Methods. Oxford: Oxford Univ. Press.

Brynard, P.A. and Hanekom, S.X. (2006) *Introduction to research in management- related fields.* 2nd edn. RSA, Pretoria: Van Schaik Hatfield Publishers.

Buchanan, T., Sainter, P., and Saunders, G. (2013) Factors affecting faculty use of learning technologies: Implications for models of technology adoption, *Journal of Computer in Higher Education*, 25(1), pp.1–11.

Butler, M., and Morgan, M. (2007) Learning challenges faced by novice programming students studying high level and low feedback concepts. *In Proceedings of Ascilite,* pp. 99-107, Singapore, 2-5 December 2007.

Byrne, B. M. (2016) *Structural Equation Modelling with AMOS: Basic Concepts, Applications, and Programming* (3rd ed.) New York: Routledge.

Castro, V., B., de, Sridharan, B., Watty, K., and Safari, M. (2020). The impact of learner engagement on performance outcomes: a longitudinal study in accounting education. *Accounting & Finance*. 61, pp. 1639-1664. doi:10.1111/acfi.12640. ISSN 1467-629X.

Cavalcante, M. T. L., Riberas, G., and Rosa, G. (2016) Fostering innovation in social work and social education degrees: multilingual environment and tools for social change, *International Journal of Educational Technology in Higher Education*, 13(1). DOI: 10.1186/s41239-016-0031-0.

Celik, H. (2018). The effects of activity-based learning on sixth grade students' achievement and attitudes towards mathematics activities. *Eurasia Journal of Mathematics, Science and Technology Education*, 14 (5), pp. 1963-1977.

Chaparro, E., A., Yuksel A, Romero, P., Bryant, S. (2005) Factors Affecting the Perceived Effectiveness of Pair Programming in Higher Education, *Proceedings 17th Workshop of the Psychology of Programming Interest Group*, pp. 5-18.

Chatti, M.A., Agustiawan, M.R. Jarke, M., Specht, M. (2010) Toward a Personal Learning Environment Framework. *International Journal of Virtual and Personal Learning Environments* 1(4), pp. 66–85.

Chee, J. (2009) *Action research project design document*. Information Packet.
Retrieved: August 20, 2019, Available at: http://www.ed369a.pbworks.com/f/ED369-AR-design.pdf.

Chen, J. and Zelinsky, A. (2003) Programming by demonstration: Coping with suboptimal teaching actions, *International Journal of Robotics Research*, 22(5), pp. 299–319.
doi: 10.1177/027836403128965088.

Chew, E., Jones, N., and Turner, D. (2008) Critical Review of the Blended Learning Models Based on Maslow's and Vygotsky's Educational Theory. *International Conference on Hybrid Learning and Education*, pp. 40-53.

Child, D., (2006) *The essentials of factor analysis*. 3rd edn. London: Continuum.

Childer, J. S. (1961) The act of discovery, *Harvard Educational Review*, 31(1), pp. 21-32.

Chung, E., Subramaniam, G. and Dass, L. C. (2020) Online Learning Readiness among University Students in Malaysia amidst COVID-19, *Asian Journal of University Education (AJUE)*, 20, pp. 46–58. Available at: https://eric.ed.gov/?id=EJ1267359.

Coates, H., James, R. and Baldwin, G. (2005) A critical examination of the effects of learning management systems on university teaching and learning, *Tertiary Education and Management*, 11(1), pp. 19-36. doi: 10.1080/13583884.2005.9967137.

Code.org. (2016) *Leaders and trend-setters all agree on one thing*. Retrieved from https://code.org/quotes.

Cohen, D.K. and Ball, D.L. (1999) *Instruction, capacity, and improvement*. *Research report series No. 43*. Philadelphia, PA: Consortium for Policy Research in Education (CPRE).

Cohen, D. L. (1990) Montessori Methods in Public Schools. *Education Digest.* 55 (9), pp. 20-23.

Cohen, L., Manion, L., and Morrison, K. (2012) *Research Methods in Education, Professional Development in Education.* London: Routledge. Available at: doi: 10.1080/19415257.2011.643130.

Cohen, L., Manion, L. and Morrison, K. (2005) *Research methods in education*. 5th edn. London: Routledge.

Collis, B. and Winnips, K. (2002) Two scenarios for productive learning environments in the workplace, *British Journal of Educational Technology*, 33 (2), pp.133-148.

Conner, M. (2020) *Introduction to Adult Learning*. Retrieved June 16, 2020, from https://marciaconner.com/resources/adult-learning/.

Cook, C. W. and Sonnenberg, C. (2014) Technology and online education: Models for change. *ASBBS E-Journal,* 10(1), pp. 43–59.

Creswell, J. W. (2008) *Three components involved in a design, Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, pp. 5–21. Available at: http://www.sagepub.com/upm-data/22780_Chapter_1.pdf.

Creswell, J. W. (2014) *Research design. Qualitative, quantitative and mixed methods approaches*, Thousand Oaks: Sage Publications, Inc., pp. 398. doi: 10.1007/s13398-014-0173-7.2.

Creswell, J.W. and Poth, C.N. (2018*) Qualitative inquiry and research design choosing among five approaches*. 4th edn. Thousand Oaks: Sage Publications, Inc.

Cripps C. J., Jacobs, B. and MacCallum, J. (2020) Solidarity and Community: Collaborative Learning in Times of Crisis. *Hu Arenas*. https://doi.org/10.1007/s42087-020-00152-4.

Cronbach, L. J. (1951) Coefficient alpha and the internal structure of tests *Psychometrika,* 16, pp. 297-334.

da Cruz, F. (2020) Columbia University Computing History - Herman Hollerith. *Columbia University*. Columbia.edu. Archived from the original on April 29, 2020. Retrieved April 25, 2021.

Dalsgaard, C. (2006) Social software: e-learning beyond learning management systems, *European Journal of Open, Distance and E-learning*, 9(2), pp. 1–7. Available at: http://www.eurodl.org/index.php?p=archives&year=2006&hal&article=228.

Daniel, J., Vázquez Cano, E., and Gisbert, M. (2015) The Future of MOOCs: Adaptive Learning or Business Model? RUSC. *Universities and Knowledge Society Journal*, 12(1), pp. 64-74. https://doi.org/10.1145/1028174.971303.

Daniel, S. J., Vázquez Cano, E. and Gisbert, M. (2015) The Future of MOOCs: Adaptive Learning or Business Model? *RUSC. Universities and Knowledge Society Journal*, 12(1), pp. 64. doi: 10.7238/rusc.v12i1.2475.

Davis, B., Carmean, C., and Wagner, E. (2009) The Evolution of the LMS: From Management to Learning, The *E-Learning Guild Research*.

Dağ, F. and Geçer, A. (2009) Relations between online learning and learning styles, *Procedia - Social and Behavioral Sciences,* 1(1), pp. 862–871. doi: 10.1016/j.sbspro.2009.01.155.

Denning, P. J., and Freeman, P. A. (2009) The Profession of IT Computing's Paradigm. *Communications of the ACM*, 52(12), 28. https://doi.org/10.1145/1610252.1610265.

Denning, P. J., Comer, D. E., Gries, D., Mulder, M. C., Tucker, A., Turner, A. J., and Young, P. R. (1989) Computing as a Discipline. *Communications of the ACM*, 32(1), pp. 9–23. https://doi.org/10.1145/63238.63239.

Derus, S.R.M. and Ali, A.Z.M., (2012) Difficulties in learning programming: views of students, In: ICCIE, 2012, *1st International Conference on Current Issues in Education.* Yogyakarta, Indonesia, 15-16 September 2012. Malaysia: National University of Malaysia.

Deterding, S., Dixon, D., Khaled, R., and Nacke, L. (2011) *From game design elements to gamefulness: defining 'gamification'*. 15th International MindTrek Conference. New York: ACM, pp. 9-15. doi:10.1145/2181037.2181040. ISBN:9781450308168. Retrieved 4 August 2019.

DeVellis, R., (2003) *Scale development: theory and applications*. Thousand Oaks: Sage Publications.

De Villiers, M.R. (2005a) E-Learning artifacts: Are they based on learning theory? Alternation, 12(1b), pp. 345-371.

De Villiers, M.R. (2005b) Interpretive research models for informatics: action research, ground theory, and the family of design and development research, *Alternation*, 12(2), pp. 10-52.

Dewey, J. (1922) *Democracy and education: an introduction to the philosophy of education*. New York: The Macmillan Company.

Dewey, J. (1960) *How we think: a restatement of the relation of reflective thinking to the educative process*. Lexington, MA: D. C. Health and Company.

Dewey, J. (1997) *Democracy and education*. New York: Free Press.

Dhawan, S. (2020) Online Learning: A Panacea in the Time of COVID-19 Crisis, *Journal of Educational Technology Systems*, 49(1), pp. 5–22. doi: 10.1177/0047239520934018.

Dietinger, T. (2003) Aspects of e-learning environments. *Unpublished doctoral dissertation*, Graz University of Technology, Austria.

Dijkstra, T.K. and Henseler, J. (2015) Consistent partial least squares path modeling. *MIS Quarterly*, (39), pp. 297-316. Available at: https://doi.org/10.25300/MISQ/2015/39.2.02

Dick, B. (2002) Postgraduate programs using action research. *The Learning Organization,* 9, pp. 159-170.

Docktor, J. L. (2009) Development and validation of a physics problem-solving assessment rubric (*Unpublished doctoral dissertation*) Physics, University of Minnesota.

Docktor, J. L., and Heller, P. (2009) Robust assessment instrument for student problem solving. In *Proceedings of the NARST 2009* annual meeting.

Dodge, B. (1998) *Active Learning on the Web*, Available at: http://edweb.sdsu.edu/people/bdodge/ActiveLearning.html. Accessed on 21st January, 2018.

Etikan, I., Musa, S. A., and Alkassim, R. S. (2016) Comparison of Convenience Sampling and Purposive Sampling, *American Journal of Theoretical and Applied Statistics*, 5, pp. 1-4. https://doi.org/10.11648/j.ajtas.20160501.11.

Fadde, P. J., and Phu V. (2014) Blended Online Learning: Benefits, Challenges and Misconceptions. In P. Lowenthal, C. S. York, and J. C. Richardson (Eds.), *Online Learning: Common Misconceptions, Benefits and Challenges*, pp. 33 – 48, New York: Nova.

Falkner. N., Sooriamurthi, R., Michalewicz, Z. (2010) Puzzle-Based Learning for Engineering and Computer Science, *Computer*, USA. 43(4), pp. 20-28, http://doi.ieeecomputersociety.org/10.1109/MC.2010.113

Fisher, L., Byrne, J. R. and Tangney, B. (2016) A 21st Century Teaching and Learning Approach to Computer Science Education: Teacher Reactions, In Zvacek, S. et al. (eds.) *Computer Supported Education*. *Cham*: *Springer International Publishing*, pp. 523–540.

Fisher, L. Byrne, J. R. and Tangney, B. (2016) Teacher experiences of learning computing using a 21st century model of computer science continuing professional development, in *8th International Conference on Computer Supported Education (CSEDU 2016) -. SCITEPRESS – Science and Technology Publications*, LDA., pp. 273–280. Available at: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84979554396&partnerID=40&md5=c9b7b9fcf4a403b84bb6d29cd4626da8.

Flores, Ò., del-Arco, I. and Silva, P. (2016) The flipped classroom model at the University: analysis based on professors and students' assessment in the educational field, International *Journal of Educational Technology in Higher Education*, 13(1) Available at: doi: 10.1186/s41239-016-0022-1.

Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., Wenderoth, M. P. (2014) Active learning increases student performance in science, engineering, and mathematics, *Proceedings of the National Academy of Sciences*, 111(23), pp. 8410–8415. doi: 10.1073/pnas.1319030111.

Freitas, F. A. and Leonard, L. J. (2011) Maslow's hierarchy of needs and student academic success, *Teaching and Learning in Nursing*. Elsevier B.V., 6(1), pp. 9-14. doi: 10.1016/j.teln.2010.07.004.

Fuegi, J. and Francis, J. (2003). Lovelace & Babbage and the Creation of the 1843 'notes'. *IEEE Annals of the History of Computing*. 25 (4), pp. 16 -19. doi:10.1109/MAHC.2003.1253887

Gal-Ezer, J., and Harel, D. (1998) What (else) should CS educators know? *Communications of the ACM*, 41(9), pp. 77–84. https://doi.org/10.1145/285070.285085

Garner, S. (2003) Learning Resources and Tools to Aid Novices Learn Programming. *Informing Science and Information Technology Education Joint Conference (INSITE)* Pori, Finland, 24-27 June 2003.

Garrison, D. R., and Anderson, T. (2003) *E-learning in the 21st century: a framework for research and practice*. London: Routledge.

Garrison, D. R., Anderson, T. and Archer, W. (2010) The first decade of the community of inquiry framework: A retrospective, *Internet and Higher Education*. Elsevier Inc., 13(1–2), pp. 5–9. doi: 10.1016/j.iheduc.2009.10.004.

Garrison, D. R., Cleveland-Innes, M. and Fung, T. S. (2010) Exploring causal relationships among teaching, cognitive and social presence: Student perceptions of the community of inquiry framework, *Internet and Higher Education.* Elsevier Inc., 13(1–2), pp. 31–36. doi: 10.1016/j.iheduc.2009.10.002.

Gilboy, M. B., Heinerichs, S., and Pazzaglia, G. (2015) Enhancing student engagement using the flipped classroom, *Journal of Nutrition Education and Behavior*, 47(1), pp. 109–114

Goel, T. (2017) *Why we need learning theories [Web log post].* Designed for Learning! Retrieved from https://edtechbooks.org/-wqZ on August 2, 2020.

Goodyear, P. and Retalis, S. (2010) *Technology-Enhanced Learning*, Sense Publishers. Available at: doi: 10.1080/14759390802383827.

Grant, C., Osanloo. A. (2014) Theoretical Framework in Dissertation Research, *Administrative Issues Journal: Connecting Education, Practice, and Research*, 4(2), pp. 12–26. doi: 10.5929/2014.4.2.9.

Greener, S. (2008) *Business research methods*. London: Ventus Publishing.

Greyling F.C and Wentzel A. (2007) Humanising education through technology: creating social presence in large classes. *South African Journal of Higher Education*, 22(4), pp 654-667.

Gros, B. (2016) The design of smart educational environments, *Smart Learning Environments*. Available at: doi: 10.1186/s40561-016-0039-x.

Guarino, N. (2005) Ontology-driven conceptual modelling,*9th East-European Conference on Advances in Databases and Information Systems* (ADBIS2005).

Guizzardi, G., Herre, H., and Wagner, G. (2002) On the general ontological foundations of conceptual modelling, In *Proceedings of 21st International Conference on Conceptual Modelling*, ER 2002, LNCS 2503, pp. 65–78, Tampere, Finland.

Hagan, D., and Markham, S. (2000) Does it help to have some programming experience before beginning a computing degree program? In *ACM SIGCSE bulletin*, Vol. 32, pp. (25–28).

Hair, J., Black, W., Babin, B. Y. A., Anderson, R., and Tatham, R. (2010) *Multivariate data analysis*. 7th edn. New Jersey: Pearson Prentice Hall.

Hair, J. F. (2010) *Multivariate data analysis: a global perspective*. Upper Saddle River, New Jersey: Person Prentice Hall.

Hair, J.F., Anderson, R.E., Babin, B.J. and Black, W.C., (2010) *Multivariate data analysis: A global perspective*. 7th edn. London: Pearson Education Limited.

Hair, J.F., Hult, G.T.M., Ringle, C.M. and Sarstedt, M. (2014) *A primer on Partial Least Squares Structural Equation Modeling*. Thousand Oaks, California: Sage.

Hair, J. F., Wolfinbarger, M., Ortinau, D. and Bush, R. (2008) *Essentials of marketing research*. New York: McGraw Hill/Irwin.

Harasim, L. (2012) *Learning Theory and Online Technologies. Taylor and Francis*, London: Routledge, pp. 11–12. ISBN: 978-0-415-99975-5, Available at: doi: 10.4324/9781315716831-5.

Harris, J., Mishra, P. and Koehler, M. (2008) Teachers Technological Pedagogical Content Knowledge and Learning Activity Types: Curriculum-Based Technology Integration Reframed, *Journal of Research on Technology in Education*, 4(2003), pp. 669–690. doi: 10.1207/s15326985ep2803_7.

Haungs, M., Clark, C., Clements, J., and Janzen, D. (2012) Improving first-year success and retention through interest-based cs0 courses. In *Proceedings of 43rd ACM technical symposium on computer science education,* pp. 589–594.

Hazzan, O., Lapidot, T. and Ragonis, N. (2014) *Guide to Teaching Computer Science: An Activity-Based Approach*. 2nd edn. Springer-Verlag London, doi: 10.1007/978-1-4471-6630-6.

Hazzan, O., Ragonis, N., and Lapidot, T. (2020) *Guide to Teaching Computer Science: An Activity-Based Approach*. 3rd edn. Springer International Publishing, doi:10.1007/978-3-030-39360-1

Heller, K., and Heller, P. (2010) *Cooperative problem solving in Physics: a user's manual.* Minnesota: National Science Foundation, University of Minnesota, and the USA Department of Education.

Henseler, J., Hubona, G. and Ray, P. A. (2016) Using PLS path modelling in new technology research: Updated guidelines, *Industrial Management and Data Systems*, 116(1), pp. 2–20. doi: 10.1108/IMDS-09-2015-0382.

Henseler, J., Ringle, C.M. and Sinkovics, R.R. (2009) The use of partial least squares path modeling in international marketing, *Advances in International Marketing*, 20, pp. 277-320.

Herington, C. and Weaven, S. (2008) Action research and reflection on student approaches to learning in large first year university classes, *The Australian Educational Researcher*, 35(3), pp. 111-134.

Horwitz, S., Rodger, S. H., Biggers, M., Binkley, D., Frantz, C. K., Gundermann, D., others (2009) Using peer-led team learning to increase participation and success of under-represented groups in introductory computer science. *ACM SIGCSE Bulletin*, 41(1), pp. 163–167.

Hu, L.T. and Bentler, P.M., (1995) *Evaluating Model Fit*. In R.H. Hoyle (ed.) *Structural equation modeling: Concepts, Issues, and Applications*, pp. 76-99. Washington, DC: Sage Publications.

Hu, M. (2003) A Case Study in Teaching Adult Students Computer Programming, *Proceedings of the 16th Annual NACCQ*, Palmerston, North- New Zealand, pp. 287-290.

Huan, T., Shehane, R. and Ali, A. (2011) Teaching computer science courses in distance learning, *Journal of Instructional Pedagogies*, 6, pp. 1–14.

Huang, R, Ma, D. and Zhang, H. (2008) Towards a design theory of blended learning curriculum, in *Hybrid Learning and Education*, pp. 66–78. doi: 10.1007/978-3-540-85170-7_6.

Huyer, S. (2015) *Is the gender gap narrowing in science and engineering*, UNESCO Global Science Report: Towards 2030, pp. 85–103. Available at: http://unesdoc.unesco.org/images/0023/002354/235406e.pdf%0Ahttp://unesdoc.unesco.org/images/0023/002354/235407e.pdf.

Hwang, G.J. (2014) Definition, framework and research issues of smart learning environments-a context-aware ubiquitous learning perspective. *Smart Learning Environ*ment 1(1), pp. 1–14

Hwang, G.J., Chu, H.C., Yin, C., Ogata, H. (2015) Transforming the educational settings: innovative designs and applications of learning technologies and learning environments. *Interactive Learning Environ*ment 23(2), pp. 127–129.

Jain, N. (2010) *Can we make our kids smarter?* Huffington Post. Retrieved November 8, 2010, from http://www.huffingtonpost.com/naveen-jain/post_1215_b_779221.html

Jain, S. and Angural, V. (2017) Use of Cronbach's alpha in Dental Research, in Jain S. and Angural V., *Medical Research Chronicle*, 4(3), pp. 285–291.

Jenkins, T. (2002) On the Difficulty of Learning to Program, *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*, Lough- borough, UK, pp.53-58.

Jenkins, T. (2002) Teaching Programming - A Journey from Teacher to Motivator, *Proceedings of 2nd Annual LTSN-ICS Conference*, pp. 65-71.

Jennifer, A. J. S. and Monfries, M. (1995) Motivation to Learn in University Students: Links with Vygotsky's Assisted Discovery, *European Conference on Educational Research*, September 14-17, 1995, Bath.

Kalanda, K. and De Villiers, M.R. (2008) Toward ICT integration in the science classroom: A comparative study of cases in Lesotho and South Africa, In J. Luca and E. Weippl eds., *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications (EDMEDIA),* Vienna, 30 June - 4 July 2008, pp. 3175-3182.

Kanaparan, G., Cullen, R., and Mason, D. (2013) Self-Efficacy and Engagement as Predictors of Student Programming Performance, In Proceedings of *17th Pacific Asia Conference on Information Systems PACIS*, pp. 282-282. JEJU Island, Korea.

Kanet, J. and M. Barut. (2004) Problem-based learning for production and operations management, *Decision Sciences Journal of Innovative Education*, 1(1), pp. 99-118.

Kanuka, H. and Garrison, D. R. (2004) Cognitive presence in online learning, *Journal of Computing in Higher Education*, 15(2), pp. 21–39. doi: 10.1007/BF02940928.

Kapp, K. (2012). *The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education*. Pfeiffer. ISBN 978-1118096345.
Kasame, T., Pachoen, K. and Manit, A. (2016) The use of engineering design concept for computer programming course: A model of blended learning environment, *Educational Research and Reviews*, 11(18), pp. 1757–1765. doi: 10.5897/ERR2016.2948.

Kay, D. and Kibble, J. (2016) Learning theories 101: application to everyday teaching and scholarship. *Advances in Physiology Education*, 40(1), pp. 17-25.

Khan, S. (2011) New Pedagogies on Teaching Science with Computer Simulations, *Journal of Science Education and Technology*, 20(3), pp. 215–232. doi: 10.1007/s10956-010-9247-2.

Kharb, P. and Samanta, P. P. (2016) Blended learning approach for teaching and learning anatomy: Students and teachers' perspective, *Journal of the Anatomical Society of India*, 65(1), pp. 43-47. doi: 10.1016/j.jasi.2016.06.001.

Kinnunen, P., and Malmi, L. (2006) Why students drop out cs1 course? Proceedings of the 2006 *International Workshop on Computing Education Research - ICER 06*. doi: 10.1145/1151588.1151604.

Kinnunen, P. and Malmi, L., (2008) CS minors in a CS1 course, In: *ICER 08, Proceedings of the 4th International Workshop on Computing Education Research*, Sydney, Australia, 06-07 September 2008. New York: ACM.

Kinshuk, Chen, N. S., Cheng, I. L., and Chew, S. W. (2016). Evolution is not enough: revolutionizing current learning environments to smart learning environments. *International Journal of Artificial Intelligence in Education*, 26, 561–581. https://doi.org/10.1 007/s40593-016-0108-x.

Kintu, M. J., Zhu, C. and Kagambe, E. (2017) Blended learning effectiveness: the relationship between student characteristics, design features and outcomes, *International Journal of Educational Technology in Higher Education*, 14(1), pp. 7 doi: 10.1186/s41239-017-0043-4.

Kirkwood, A. and Price, L. (2016) Technology-enhanced learning and teaching in higher education: what is enhanced and how do we know? A critical literature review, *Learning, Media and Technology*, 9884(May), pp. 1-44. doi: 10.1080/17439884.2014.770404.

Klašnja-Milićević, A. et al. (2017) Enhancing e-learning systems with personalized recommendation based on collaborative tagging techniques, *Applied Intelligence*, pp. 1–17. doi: 10.1007/s10489-017-1051-8.

Kosterelioglu, I. and Yapici, M. (2016). The effects of activity based learning process on prospective teachers' perceptions of constructivist learning environment. *International Journal of Human Sciences*, Vol. 13 (1), pp. 1342-1354.

Kundi, G. M. and Nawaz, A. (2010) From Objectivism to Social Constructivism: The Impacts of Information and Communication Technologies (ICTs) On Higher Education, *Journal of Science and Technology Education Research*, 1(2), pp. 30–36. Available at: http://www.academicjournals.org/JSTER.

Lang, C., McKay, J. and Lewis, S. (2007) Seven factors that influence ICT student achievement, ITiCSE 2007: 12th Annual Conference on Innovation and Technology in Computer Science Education - *Inclusive Education in Computer Science*, pp. 221–225. doi: 10.1145/1268784.1268849.

Lasse, H. (2015) *Gameful Approaches for Computer Science Education*. Available at: http://urn.fi/URN:ISBN:978-952-60-6173-3.

Latchem, C. and Jung, I. (2009) *Distance and Blended Learning in Asia: The Open and Flexible Learning Series*. Available at: http://gen.lib.rus.ec/book/index.php?md5=C1134DE411CDA0995CAF5CB2C1D0BEEA.

Leung, H., Li, F., Lau, and R., Li, Q. (eds.) Advances in Web Based Learning, Springer: Lecture Notes in Computer Science 4823, pp. 123-135. Available at http://www.springerlink.com/content/b35336204387v576.

Laurie, W. (2001). *Integrating pair programming into a software development process*. 14th Conference on Software Engineering Education and Training. Charlotte. pp. 27–36. doi:10.1109/CSEE.2001.913816. ISBN 0-7695-1059-0.

Lewis, J. and Loftus, W. (2009) *Java Software Solutions: Foundations of Program Design*, 6th edn., Addison Wesley, Boston, MA. Linn.

Li, E. Y. (1990) Software testing in a system development process: A life cycle perspective. *Journal of Systems Management*, 41(8):23-31.

Li, M. (2010) Research on the pedagogies of computer science, in ICCSE 2010 - 5th *International Conference on Computer Science and Education*, Final Program and Book of Abstracts, pp. 266–270. doi: 10.1109/ICCSE.2010.5593637.

Li L., Wang C. (2013) *Dynamic Analysis and Debugging of Binary Code for Security Applications*. In: Legay A., Bensalem S. (eds) Runtime Verification. RV 2013. Lecture Notes in Computer Science, Vol. 8174. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-40787-1_31.

Lincoln, Y.S. and Guba, E.G. (1985) *Naturalistic inquiry. Beverly Hills*, California: Sage Publications.

Lindell, M.K. and Whitney, D.J., (2001) Accounting for common method variance in cross-sectional research designs, *Journal of Applied Psychology*, 86(1), pp.114-121.
Available at: https://doi.org/10.1037/0021-9010.86.1.114.

Linn, M. C. (2009) Learning to teach computer programming, in *Proceedings of the fifth international workshop on Computing education research workshop* - ICER09, pp. 1–2.
doi: 10.1145/1584322.1584324.

Lister, R., Berglund, A., Box, I., Cope, C., Pears, A., Avram, C., . . . others (2007) Differing ways that computing academics understand teaching. *In Proceedings of the ninth Australasian conference on computing education*, 66, pp. 97–106

Lister, R., Seppälä, O., Simon, B., Thomas, L., Adams, E. S., Fitzgerald, S. et al. (2004) A multi-national study of reading and tracing skills in novice programmers. *ACM SIGCSE Bulletin*, 36 (4), 119–150. doi: 10.1145/1041624.1041673.

Lye, S. Y., and Koh, J. H. L. (2014) Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, pp. 51–61. https://doi.org/10.1016/j.chb.2014.09.012.

López-Cruz, O. Mora, A. L., S., Mauricio E., Diana, L., (2017) Teaching computer programming as knowledge transfer: Some impacts on software engineering productivity, in *Advances in Intelligent Systems and Computing*, pp. 145–154. doi: 10.1007/978-3-319-48523-2_14.

Mackay, F. (2010) *Business schools respond to demand for use of social media*. The New York Times. Available at: http://www.nytimes.com/2010/03/31/education/31ihtriedmba.html?scp=1&sq=&st=nyt.

Malmi, L., Sheard, J., Kinnunen, P., Sinclair, J. (2019) Computing education theories: What are they and how are they used?, *ACM Inroads*, 11(1), pp. 54–64. doi: 10.1145/3381889.

Margaryan, A., Collis, B. and Cooke, A. (2004) Activity-based blended learning, *Human Resource Development International*, 7(2), pp. 265–274. doi: 10.1080/1367886042000338227.

Margolis, E., and Laurence, S. (Eds.) (1999*) Concepts: Core readings*. The MIT Press.

Margolis, J., and Fisher, A. (2002) *Unlocking the clubhouse: Women in computing*. Cambridge: Massachusetts Institute of Technology.

Matarirano, O., Jere, N., Sibanda, H. and Panicker, M. (2021) Antecedents of Blackboard Adoption by Lecturers at a South African Higher Education Institution - Extending GETAMEL. *International Journal of Emerging Technologies in Learning*, 16(1), pp. 60-79.

Mavuso, N. and Jere, N. (2020). An Innovative Blended Learning Model for Information Technology Courses for a South African University, *2020 IST-Africa Conference*, pp. 1-9.

McLeod, S. (2007) HCC Certificate in Counselling Skills Maslow's Hierarchy of Needs, *Simple Psychology*, (December), pp. 4. doi: 10.1016/B978-0-88415-752-6.50250-2.

Meijers, A. W. M., Gabbay, D. M., Thagard, P., and Woods J (2009) *Philosophy of Technology and Engineering Sciences*, North Holland, Amsterdam.

Menard, S., (1995) A developmental test of Mertonian anomie theory, *Journal of Research in Crime and Delinquency*, 32(2), pp. 136-174.

Merrick, K., E. (2010) An Empirical Evaluation of Puzzle-Based Learning as an Interest Approach for Teaching Introductory, *Computer Science, IEEE Transactions on Education*, 53(4), pp. 677-680.

Merrill, D. (2003) First principles of instruction, *Educational Technology Research and Development* 50(3): pp.43 – 59.

Meyer, K., A. (2006) The Closing of the US Open University, *Educause Quarterly*, 29(2), pp. 5-7.

Millheim, K. (2012) Toward a Better Experience: Examining Student Needs in the Online Classroom through Maslow's Hierarchy of Needs Model, *Journal of Online Learning and Teaching*, 8(2), pp. 159–171. doi: 10.1080/713688409.

Mishra, P. and Koehler, M. J. (2006) Technological pedagogical content knowledge: A framework for teacher knowledge, *Teachers College Record*, pp. 1017–1054.
doi: 10.1111/j.1467-9620.2006.00684.x.

Mohorovičić, S., Tijan, E. (2010) New Technologies in Teaching University Level Programming, *Proceedings of the 33rd International Convention on information and communication technology, electronic and microelectronics (MIPRO 2010),* Vol. IV. CE, Croatia, pp. 268-272.

Moorthy, G. T. and Arulsamy, S. (2014) Understanding the paradigm shift in teaching and learning, *International Journal of Social Science*, 3(4), pp. 443–446.

Morse, J., M. (1991) Approaches to qualitative-quantitative methodological triangulation, *Nursing Research*, 14, pp.120-124.

Moule, P. (2007) Challenging the five-stage model for e-learning: a new approach, *Research in Learning Technology*, 15(1) doi: 10.3402/rlt.v15i1.10911.

MTH (2007) Birmingham to buy 15K laptops from OLPC, *The Journal of New England Technology*. Retrieved 10 Sept 2019 from: http://morganya.org/research/warschauer-olpc-birmingham.pdf.

Myrup, J. (2017) Aalborg Universitet Blended Learning and Problem Based Learning in a multinational and multidisciplinary setting: Aalborg Universitet PBL, *Social Progress and Sustainability*, Guerra.

Mödritscher, F. (2006) e-Learning Theories in Practice: A Comparison of three Methods, *Journal of Universal Science and Technology of Learning*, 2(2) pp. 3-18.

Nazarenko, A. L. (2015) Blended Learning vs Traditional Learning: What Works? (A Case Study Research), *Procedia - Social and Behavioral Sciences*. Elsevier B.V., pp. 77–82. doi: 10.1016/j.sbspro.2015.08.018.

Nel, L. (2017). Students as collaborators in creating meaningful learning experiences in technology-enhanced classrooms: An engaged scholarship approach. *British Journal of Education Technology*, 48, pp. 1131-1142.

Nel, L., Wilkinson, A. (2006) Enhancing Collaborative Learning in a Blended Learning Environment: Applying a Process Planning Model. *Systematic Practice and Action Research,* 19, pp. 553–576. https://doi.org/10.1007/s11213-006-9043-3

Ngwenya S., Futcher L. (2020) A Framework for Integrating Secure Coding Principles into Undergraduate Programming Curricula. In: Tait B., Kroeze J., Gruner S. (eds) ICT Education. SACLA 2019. *Communications in Computer and Information Science*, 1136. Springer, Cham. https://doi.org/10.1007/978-3-030-35629-3_4.

Nkwo, M. S., Orji, R., and Ugah, J. (2021) Insider Perspectives of Human-Computer Interaction for Development Research: Opportunities and Challenges. In *3rd African Human-Computer Interaction Conference* (AfriCHI 2021), March 08–12, 2021, Maputo, Mozambique. ACM, New York, NY, USA. https://doi.org/10.1145/3448696.3448709.

Nouri, J. (2016) The flipped classroom: for active, effective and increased learning – especially for low achievers, *International Journal of Educational Technology in Higher Education*, 13(1) doi: 10.1186/s41239-016-0032-z.

Nunnally, J.C. (1978) *Psychometric theory*. 2nd Edition, McGraw-Hill, New York.

Nunnally, J.C. and Bernstein, I.H. (1994) The Assessment of Reliability. *Psychometric Theory*, 3, pp. 248-292.

Nuutila, E., Torma, S., Malmi, L. (2005) PBL and Computer Programming – The Seven Steps Method with Adaptations, *Computer Science Education*, 15 (2), pp. 123-142.

Okaz, A. A. (2015) Integrating Blended Learning in Higher Education, Procedia - *Social and Behavioral Sciences*. Elsevier B.V., 186, pp. 600–604. doi: 10.1016/j.sbspro.2015.04.086.

Oliva, P.F. and Gordon, W.R., (2012) *Developing the curriculum*. 8th edn. Boston: Allyn and Bacon.

OLPC (2019) *One Laptop Per Child*. Retrieved 10 September 2019 from: http://laptop.org/en/vision/index.shtml.

Olsson, M. and Mozelius, P. (2016) On Design of Online Learning Environments for Programming Education, *Proceedings of the 15th European Conference on E-Learning*, Jarmila Novotna, Antonin Jancarik, Academic Conferences Limited, 2016, pp. 533–539.

OU (2008) *The UK Open University*. Retrieved 10 Sept 2019 from: http://www.open.ac.uk.

Overmars, M. (2004) Teaching computer science through game design, *Computer*, 37(4), pp. 81-84. doi: 10.1109/MC.2004.1297314.

Pallant, J. (2016) *SPSS Survival Manual: A Step by Step Guide to Data Analysis Using SPSS Program* (6th ed.) London, UK: McGraw-Hill Education.

Pallant, J., (2005) *SPSS survival manual, Crowss Nest*, New South Wales: Allenand Unwin.

Pallant, J., (2010) *A step by step guide to data analysis using SPSS. Berkshire*, UK: McGraw-Hill Education.

Pallant, J., (2010) *A step by step guide to data analysis using SPSS*. Berkshire: McGraw-Hill Education.

Pan, H. (2020) A Glimpse of University Students' Family Life Amidst the COVID-19 Virus, *Journal of Loss and Trauma*. Routledge, 25(6–7), pp. 594–597. doi: 10.1080/15325024.2020.1750194.

Papp-Varga, Z., Szlavi, P., Zsak, O. L. (2008) ICT teaching methods Programming languages, *Annales Mathematicae et Informaticae*, 35, pp. 163-172.

Park, H.M., (2008) Univariate analysis and normality test using SAS, Stata, and SPSS. Working Paper. *The University Information Technology Services (UITS) Center for Statistical and Mathematical Computing*, Indiana University. http://www.indiana.edu/~statmath/stat/all/normality/index.html.

Park, H.M., (2015) *Univariate analysis and normality test using SAS, Stata, and SPSS.* Available at: https://www.semanticscholar.org/paper/Univariate-Analysis-and-Normality-Test-Using-SAS%2C-Park/15f7d9556bbb0df6f0c01bfd18dee2f872c82c72.

Pavla, S., Hana, V. and Jan, V. (2015) Blended Learning: Promising Strategic Alternative in Higher Education, *Procedia - Social and Behavioral Sciences.* Elsevier B.V., 171, pp. 1245–1254. doi: 10.1016/j.sbspro.2015.01.238.

Petersen, A., Craig, M., Campbell, J., and Tafliovich, A. (2016) Revisiting why students drop cs1. *Proceedings of the 16th Koli Calling International Conference on Computing Education Research - Koli Calling 16.* doi: 10.1145/2999541.2999552.

Phillips, R., Maor, D, Cumming-Potvin., Roberts, P., Herrington, J., Preston, G. and Moore, E. (2011) *Learning analytics and study behaviour: A pilot study*. In G. Williams, P. Statham, N. Brownand B. Cleland (Eds.), Changing Demands, Changing Directions. *Proceedings Ascilite Hobart 2011*. (pp.997-1007).

Piaget, J. (1983) Piaget's theory. P. Mussen (ed.) *Handbook of Child Psychology*. 4th edn. Vol. 1. New York: Wiley.

Pilkington, C. and Pretorius, L. (2015) A Conceptual Model of the Research Methodology Domain with a Focus on Computing Fields of Study, in *7th International Conference on Knowledge Engineering and Ontology Development*, 2(Ic3k), pp. 978–989.

Pinar, W.F., (2012) *What is curriculum theory*? 2nd edn. New York: Routledge.

Pintrich, P. R. (2000) The role of goal orientation in self-regulated learning. In *M. Boekaerts, P. R. Pintrich, and M. Zeidner (Eds.), Handbook of self-regulation*, Academic Press, pp. 451–502. Available at https://doi.org/10.1016/B978-012109890-2/50043-3.

Podsakoff, P.M., MacKenzie, S.B., Lee, J.Y. and Podsakoff, N.P., (2003) Common method biases in behavioral research: a critical review of the literature and recommended remedies, *Journal of applied psychology*, 88(5), pp. 879.

Poon, J. (2014) A cross-country comparison on the use of blended learning in property education, *Property Management*, 32(2), pp. 154–175. doi: 10.1108/PM-04-2013-0026.

Poster, M. (2005) *History in the Digital Domain*, In Land, R. and Bayne, S. (eds.) The *University in the Information Age*. London: Routledge Falmer.

Powers, K., Gross, P., Cooper, S., McNally, M., Goldman, K.J., Proulx, V. and Carlisle, M., (2006) Tools for teaching introductory programming: what works? *ACM SIGCSE Bulletin*, 38 (1), pp. 560-561.

Price, B., Mayfield, P., McFadden, A. and Marsh, G. (2000-2001) *Collaborative teaching: Special education for inclusive classrooms*. Parrot Publishing, L. L. C.

QAA (2018) *The Quality Assurance Agency for Higher Education*, UK
Retrieved 10 April 2019 from: https://www.qaa.ac.uk/quality-code/UK-Quality-Code-for-Higher-Education-2013-18.

Quin, Y. (2012). Impact of active learning in the classroom on academic achievement and motivation. *Innovation Magazine*, 11 (1), pp. 51-53. Retrieved on 29 May 2019.

Quinn, C. (2009) Social networking: Bridging formal and informal learning [online article]. *Learning Solutions*. Available at: http://www.learningsolutionsmag.com/articles/57/social-networking-bridging-formal-and- informal-learning.

Radošević, D., Lovrenčić, A., Orehovački, T (2009) New Approaches and Tools in Teaching Programming, *Proceedings of the 20th Central European Conference on Information and Intelligent Systems*, Varaždin, Croatia, pp. 49-57.

Rahmat, M., Shahrani, S., Latih, R., Yatim, N. F. M., Zainal, N. F. A., Rahman, R. A. (2012) Major Problems in Basic Programming that Influence Student Performance, *Procedia - Social and Behavioral Sciences*, 59, pp. 287–296. doi: 10.1016/j.sbspro.2012.09.277.

Reek, M., M. (1995) A top-down approach to teaching programming, *Proceedings of the twenty-sixth SIGCSE technical symposium on Computer science education*, New York, USA, pp. 6-9.

Robins, A., Rountree, J. and Rountree, N. (2003) Learning and teaching programming: A review and discussion, *Computer Science Education*, 13(2), pp. 137-172.

ISTE (2020), *ISTE Standards for Students*, Accessed on 20th September, 2020. Available at https://iste.org/standards/for-students.

Rochefort, B. N. R. (2011) Connecting Instruction to Connected Technologies- Why Bother? An instructional Designer's Perspective*, Revista de Universidad y Sociedad del Conocimiento (RUSC)*, 8(1), pp. 217–232.

Roehl, A., Reddy, S. L., and Shannon, G. J. (2013) The flipped classroom: an opportunity to engage millennial students through active learning strategies. *Journal of Family and Consumer Sciences*, 105(2), pp. 44-49.

Roffe, I. (2002) E-learning: engagement, enhancement and execution, *Quality Assurance in Education*, 10(1), pp. 40–50. doi: 10.1108/09684880210416102.

Rountree, N., Rountree, J., Robins, A., and Hannah, R., (2004) Interacting factors that predict success and failure in a CS1 course, *SIGCSE Bulletin* (Association for Computing Machinery, Special Interest Group on Computer Science Education), 36(4), pp. 101–104. doi: 10.1145/1041624.1041669.

Saeli, M., Perrenet, J., Jochems, W. M., and Zwaneveld, B. (2011) Teaching programming in secondary school: a pedagogical content knowledge perspective. *Informatics in Education - An International Journal*, 10(1), pp. 73-88.

Salkind, N. J. (2018) *Exploring research. Harlow, England*: Pearson, Available at http://lib.myilibrary.com?id=990569.

Sarpong, K. A., Arthur, J.K. and Owusu-Amoako, Y. P. (2013) Causes of Failure of Students in Computer Programming Courses: The Teacher – Learner Perspective, *International Journal of Computer Applications*, 77(12), pp. 27–32. doi: 10.5120/13448-1311.

Sasseville, B. (2004) Integrating Information and Communication Technology in the Classroom: A Comparative Discourse Analysis. *Canadian Journal of Learning Technol*ogy, 30(2), Spring. Retrieved on May 14, 2021, from http://www.cjlt.ca/.

Saunders, M., Lewis, P. and Thornhill, A. (2007) *Research methods for business students*. 4th edn. Edinburgh Gate, Harlow: Prentice Hall.

Saunders, M., Lewis, P. and Thornhill, A. (2012) *Research methods for business students*. Harlow: Pearson Education Limited.

Saunders, M., Lewis, P. and Thornhill, A. (2014) *Research methods for business students*. Harlow, England: Pearson Education Limited.

Schindler, L. A., Burkholder, G. J., Morad, O. A., and Marsh, C. (2017) Computer-based technology and student engagement: a critical review of the literature, *International Journal of Educational Technology in Higher Education*, 14(1) doi: 10.1186/s41239-017-0063-0.

Schunk, D., (2012) *Learning Theories - An Educational Perspective*, 6th ed., Pearson.

Schunk, D. H. (1989) *Self-efficacy and achievement behaviors*. Educational Psychology Review, 1, 173-208.

Sekaran, U., and Bougie, R. (2010) *Research methods for business: a skill-building approach* 5th edn. Haddington: John Wileyand Sons.

Shah, I. and Rahat, T. (2014) Effect of activity based teaching method in science. International *Journal of Humanities and Management Sciences*, 2 (1), pp. 39-41.

Shapiro, S.S. and Wilk, M.B., (1965) An analysis of variance test for normality (complete samples), *Biometrika*, 52(3/4), pp. 591-611.

Sharmin, S. (2018) And that's how it all began: A qualitative exploration of factors of motivation in early experiences with learning computer science. *EdMedia: World Conference on Educational Media and Technology*.

Sharmin, S. (2020) *The Impact of Open-ended Exercises on Student Experiences in Introductory Computer Science, Doctoral Thesis*, University of Toronto, Available at http://hdl.handle.net/1807/101348.

Schmidt, H. G., Rotgans, J. I., and Yew, E. H.J., (2011). The process of problem-based learning: What works and why. *Medical Education*. 45 (8), pp. 792-806. doi:10.1111/j.1365-2923.2011.04035.x

Shatz, I. (2015). *Using Gamification and Gaming in Order to Promote Risk Taking in the Language Learning Process*. MEITAL National Conference. Haifa, Israel: Technion. pp. 227–232. Retrieved 4 August 2019.

Sheard, J, A Carbone, Lister, R., Simon, B., Thompson, E., and Whalley, J., (2008) Going SOLO to assess novice programmers. *13th Annual Conference on Innovation and Technology in Computer Science Education* (ITiCSE 2008), Madrid, Spain, pp. 209-213.

Sheard, J., Simon, A., Carbone, D., Chinn, M-J., Laakso, T., Clear, M de Raadt, Souza, D., Harland, J., Lister, R., Philpott, A., and Warburton, G. (2011) Exploring programming assessment instruments: a classification scheme for examination questions. *Seventh International Computing Education Research Workshop (ICER 2011)*, Providence, RI, USA, 33-38.

Sheard, J, Simon, M., Hamilton, and J Lönnberg (2009) Analysis of research into the teaching and learning of programming. *Fifth International Computing Education Research Workshop* (ICER 2009), Berkeley, CA, USA, pp. 93-104.

Shuell, TJ. Cognitive Conceptions of Learning (1986) *Review of Educational Research.* 56(4):411-436. doi:10.3102/00346543056004411.

Silberman, M. L. (1996) *Active learning: 101 strategies to teach any subject*, Boston: Allyn and Bacon, pp. 2.

Simon, J., Sheard, J. A., Carbone, D. Chinn, M-J., Laakso, T., Clear, M. de Raadt, D. D., J. Harland, Lister, R., Philpott, A. and Warburton, G. (2011) Introductory programming: Examining the exams, *Conferences in Research and Practice in Information Technology Series*, 123, pp. 61–70.

Singal, N., Pedder, D., Malathy, D., Shanmugam, M., Manickavasagam, S. & Govindarasan, M. (2018). Insights from within activity based learning (ABL) classrooms in Tamil Nadu, India: 59 Teachers perspectives and practices. *International Journal of Educational Development*, Vol. 60, pp. 165-171.

Skinner, B.F. (1974) *About behaviorism*, Knopf, New York.

Skinner, B. F. (1968) *The technology of teaching. Englewood Cliffs*, New Jersey: Prentice-Hall.

Slavin, R. E. (2003) Educational Psychology. Boston: Allyn and Bacon. Students as related to achievement, gender and age. Inquiry, 9(1) *Students Journal of Educational Psychology*, 80(2), pp. 210-216.

Smith, R. B., Ungar, D. (1995) Programming as an Experience: *The Inspiration for Self, Object-Oriented Programming,9th European Conference*, Denmark, 952, pp. 303-330.

Smith G, Fidge C. (2008) On the efficacy of pre-recorded lectures for teaching introductory programming, *Proceedings of the tenth conference on Australasian computing education*, 78, pp. 129-136.

Soiferman, K. (2010) *Compare and contrast check on the title of document Inductive and deductive research approaches.* Ph.D. Thesis. University of Manitoba.

Soper, D.S., (2018) *A-priori sample size calculator for multiple regression* [Software]. Available from https://www. danielsoper.com/statcalc.

Spector, J.M (2014) Conceptualizing the emerging field of smart learning environments. *Smart Learning Environ*ment 1(1), pp. 2–10.

Stair, R., Reynolds, G. (2003) *Principles of Information Systems*, Course Technology, USA,

Subban, P. K., and Round, P. N. (2015) Differentiated Instruction at Work. Reinforcing the art of classroom observation through the creation of a checklist for beginning and pre-service teachers. *Australian Journal of Teacher Education*, 40(5) http://dx.doi.org/10.14221/ajte.2015v40n5.7.

Subramaniam, K. (2007) Teachers Mindsets and the Integration of Computer Technology, *British Journal of Educational Technology*, 38(6), pp. 1056-1071.

Suhendi, A. and. Purwarno, P. (2018) Constructivist Learning Theory: The Contribution to Foreign Language Learning and Teaching, *KnE Social Sciences*, 3(4), pp. 87. doi: 10.18502/kss.v3i4.1921.

Sung K, Hillyard C, Angotti R L, Panitz M W, Goldstein D S, Nordlinger J (2010) Game-Themed Programming Assignment Modules: A Pathway for Gradual Integration of Gaming Context into Existing Introductory Programming Courses, *IEEE Transactions on Education*, PP(99), pp. 1-12.

Szeto, E. (2015b) Community of Inquiry as an instructional approach: What effects of teaching, social and cognitive presences are there in blended synchronous learning and teaching? *Computers and Education*. Pergamon, 81, pp. 191–201. doi: 10.1016/J.COMPEDU.2014.10.015.

Tafliovich, A., Campbell, J., and Petersen, A. (2013) A student perspective on prior experience in cs1. In *Proceeding of the 44th ACM technical symposium on computer science education* pp. 239–244.

Taylor, J. (2009) Keynotes of Learning in Digital Worlds: What are we talking about? *Proceedings of the Computer-aided Learning* (CAL09*) – Learning in Digital Worlds*, 23-25th March, Brighton.

Taylor, S., J., and Bogdan, R. (1998) *Introduction to qualitative research methods: a guidebook and resource.* New York: John Wileyand Son.

Taylor-Powell, E., and Renner, M. (2003) *Analyzing qualitative data*. Retrieved from website: http://learningstore.uwex.edu/assets/pdfs/g3658-12.pdf.

Techbeamers (2019) *Software Development Life-cycle*, Accessed on 10th September, 2019. Available at: https://www.techbeamers.com/software-development-life-cycle-sdlc/#phases.

The Joint Task Force on Computing Curricula. (2013) Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. *ACM Technical Report 2013*. https://doi.org/10.1145/2534860.

Thomson, S. (2016) *Mark Zuckerberg's advice to girls: be nerds*. Retrieved from https://www.weforum.org/agenda/2016/01/mark-zuckerbergs-advice-to-girls-be- nerds.

Tie, H. H., Umar, I. N. (2010) The Impact of Learning Styles and Instructional Methods on Students Recall and Retention in Programming Education, *Proceedings of the 18th International Conference on Computers in Education*, Putrajaya, Malaysia, pp. 191-195.

Tiku, N. (2014) *How to get girls into coding*. Retrieved from http://www.nytimes.com/2014/06/01/opinion/sunday/how-to-get-girls-into- coding.html.

Tillmann, N. et al. (2013) Teaching and learning programming and software engineering via interactive gaming, *Proceedings - International Conference on Software Engineering*, pp. 1117–1126. doi: 10.1109/ICSE.2013.6606662.

Tirlea, L., Muir, S., Elphinstone, B. (2018) The use of Socrative in promoting classroom engagement: a qualitative investigation, *International Association for Statistical Education*, 10, pp. 1–5.

Titterton, N., Lewis, C. M., and Clancy, M. J. (2010) Experiences with lab-centric instruction. *Computer Science Education*, 20(2), 79-102.

Turban, E., Mclean, E., Wetherbe, J., Bolloju, N. and Davison, R., (2002) *Information technology for management: transforming business in the digital economy*. 3rd edn. New Jersey: John Wileyand Sons.

van Bruggen, J. (2005) Theory and practice of online learning. *British Journal of Educational Technology*, 36(1), pp. 111–112. doi: 10.1111/j.1467-8535.2005.00445_1.x.

VanDeGrift, T. (2007) Encouraging creativity in introductory computer science program- ming assignments. *Proceedings of the American Society for Engineering Education Conference.*

Voronina, M. V., Moroz, O. N, Sudarikov, A. E. (2017) Systematic review and results of the experiment of a flipped learning model for the courses of descriptive geometry, engineering and computer graphics, computer geometry, *Eurasia Journal of Mathematics, Science and Technology Education*, 13(8), pp. 4831-4845. doi: 10.12973/eurasia.2017.00967a.

Vygotsky, L. S. (1962) *Thought and language*. Cambridge, Massachusetts: MIT Press.

Walker, R. Voce, J. and Ahmed, J. (2012) Survey of Technology Enhanced Learning for higher education in the UK. *Universities and Colleges Information Systems Association,* pp. 2.

Warschauer, M., Cotten, S. R., and Ames, M. G. (2012) One laptop per child Birmingham: Case study of a radical experiment, *International Journal of Learning*, 3(2), pp. 61-76. doi: 10.1162/IJLM.

Watson, C. F. (2014) Failure Rates in Introductory Programming Revisited. *Proceedings of the 2014 Conference on Innovation and Technology in Computer Science Education* (ITiCSE '14) (pp. 39-44) New York: ACM.

Wiedenbeck, S. and Labelle, D. and Kain, V.N.R., (2004) Factors affecting course outcomes in introductory programming, In: *Proceedings of the 16th Annual Workshop of the Psychology* *of Programming Interest Group*. Carlow, Ireland, 5-7 April 2004. UK: Psychology of Programming Interest Group.

Wilcox, C., and Lionelle, A. (2018) Quantifying the benefits of prior programming experience in an introductory computer science course. *In Proceedings of the 49th ACM technical symposium on computer science education* pp. 80–85.

Williams, B., Onsman, A. and Brown, T., (2010) Exploratory factor analysis: A five-step guide for novices, *Australasian Journal of Paramedicine*, 8(3).

Wing, J. M. (2006) Computational Thinking. *Communications of the ACM*, 49(3), pp. 33. https://doi.org/10.1145/1118178.1118215.

Wing, J. M. (2008) Computational Thinking and Thinking About Computing. *Philosophical Transactions. Series A, Mathematical, Physical, and Engineering Sciences*, 366(1881), pp. 3717–3725. https://doi.org/10.1098/rsta.2008.0118.

Wikipedia (2021). *Activity-based learning in India - Wikipedia*. [online], Available at: https://en.wikipedia.org/wiki/Activity-based_learning_in_India#:~:text=Activity%2Dbased%20learning%20or%20ABL,hands%2Don%20experiments%20and%20activities. Accessed on 8 March 2021.

Woolfolk, A. (2007) *Educational Psychology*. 10th Edition, Allyn and Bacon, Boston.

Woolfolk H., A., Davis, H. A. and Anderman, E. M. (2013) Theories of Learning and Teaching in TIP, *Theory into Practice*, 52 (SUPPL 1), pp. 9-21. doi: 10.1080/00405841.2013.795437.

Wu, Y. (2006) Applying a hybrid problem-based learning method to the teaching of computer programming, The *China Papers*, Issue 6, pp. 63- 66.

Wulf, T. (2005) Constructivist approaches for teaching computer programming, *SIGITE 05: Proceedings of the 6th Conference on Information Technology Education*, pp. 245–248. doi: http://doi.acm.org/10.1145/1095714.1095771.

Yacob, A. and Saman, M.Y.M., (2012) Assessing level of motivation in learning programming among engineering students, In *ICIA 2012, The International Conference on Informatics and Applications,* Kuala Terengganu, Malaysia 3-5 June 2012. USA: The Society of Digital Information and Wireless Communication.

Yoneyama, Y., Matsushita, K., Mackin, K., J., Ohshiro, M., Yamasaki, K., Nunohiro, E. (2008) Puzzle Based Programming Learning Support System with Learning History Management, *Proceedings of the 16th International Conference on Computers in Education,* pp. 623-627.

Zacharis, N. (2009) Evaluating the Effects of Virtual Pair Programming on Student's Achievement and Satisfaction, *International Journal of Emerging Technologies in Learning* (IJET), 4(3), pp. 34-39.

Zacharis, N. Z. (2011) Measuring the Effects of Virtual Pair Programming in an Introductory Programming Java Course, *IEEE Transactions on Education*, 54(1), pp. 168-170.

Zingaro, D., (2014) Peer instruction contributes to self-efficacy in CS1, *SIGCSE '14: Proceedings of the 45th ACM technical symposium on Computer science education*, pp. 373–378. Accessed from https://doi.org/10.1145/2538862.2538878.

Appendix A: Ethical Clearance

UNISA university of south africa

UNISA COLLEGE OF SCIENCE, ENGINEERING AND TECHNOLOGY'S (CSET) RESEARCH AND ETHICS COMMITTEE

02 September 2019

Ref #: 029/EF/2019/CSET_SOC
Name: Mr Emmanuel Freeman
Student #: 60870540
Staff #:

Dear Mr Emmanuel Freeman

Decision: Ethics Approval for 5 years (Humans involved)

Researchers: Mr Emmanuel Freeman, Ghana Technology University College, PMB 100, Accra-North, Ghana; 60870540@mylife.unisa.ac.za, +233 2459 79789, +233 5015 61924

Project Leader(s): Prof Ian Douglas Sanders, sandeid@unisa.ac.za, +27 11 471 2858
Dr Bester Chimbo, chimbb@unisa.ac.za, +27 11 670 9105

Working title of Research:

A Framework for an Activity-Based Computer Programming Instruction in a Blended Learning Environment of Higher Education Institutions in Ghana

Qualification: PhD in Computer Science

Thank you for the application for research ethics clearance by the Unisa College of Science, Engineering and Technology's (CSET) Research and Ethics Committee for the above mentioned research. Ethics approval is granted for a period of five years, from 02 September 2019 to 02 September 2024.

1. The researcher will ensure that the research project adheres to the values and principles expressed in the UNISA Policy on Research Ethics.
2. Any adverse circumstance arising in the undertaking of the research project that is relevant to the ethicality of the study, as well as changes in the methodology, should be communicated in writing to the Unisa College of Science, Engineering and Technology's (CSET) Research and Ethics Committee. An amended application could

University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa
Telephone: +27 12 429 3111 Facsimile: +27 12 429 4150
www.unisa.ac.za

be requested if there are substantial changes from the existing proposal, especially if those changes affect any of the study-related risks for the research participants.

3. The researcher(s) will conduct the study according to the methods and procedures set out in the approved application.
4. Any changes that can affect the study-related risks for the research participants, particularly in terms of assurances made with regards to the protection of participants' privacy and the confidentiality of the data, should be reported to the Committee in writing, accompanied by a progress report.
5. The researcher will ensure that the research project adheres to any applicable national legislation, professional codes of conduct, institutional guidelines and scientific standards relevant to the specific field of study. Adherence to the following South African legislation is important, if applicable: Protection of Personal Information Act, no 4 of 2013; Children's act no 38 of 2005 and the National Health Act, no 61 of 2003.
6. Only de-identified research data may be used for secondary research purposes in future on condition that the research objectives are similar to those of the original research. Secondary use of identifiable human research data requires additional ethics clearance.
7. The signed permissions from Ghana Technology University College (GTUC), Accra Institute of Technology (AIT) and Valley View University should be obtained prior to commencing field work.
8. No field work activities may continue after the expiry date (02 September 2024).
9. Submission of a completed research ethics progress report will constitute an application for renewal of Ethics Research Committee approval.

Note:

The reference number 029/EF/2019/CSET_SOC should be clearly indicated on all forms of communication with the intended research participants, as well as with the Unisa College of Science, Engineering and Technology's (CSET) Research and Ethics Committee.

Yours sincerely

Dr Danie Bisschoff

Sub-Chair: Ethics Sub-Committee SoC, College of Science, Engineering and Technology (CSET)

Prof I. Osunmakinde
DR KATUMBA
Director: School of Computing, CSET

Prof B Mamba
Executive Dean: CSET

University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa
Telephone: +27 12 429 3111 Facsimile: +27 12 429 4150
www.unisa.ac.za

Appendix B: Questionnaire for Respondents (Students) – First Semester

Dear Student,

This questionnaire aims at soliciting your views and experiences on Activity-Based Computer Programming Instruction in a Blended Learning Environment of Higher Education Institutions in Ghana. This research is intended for solely academic purposes, in view of that please indicate your candid knowledge and responses for the questions. Confidentiality and anonymity of your responses are assured and will be used only for writing my PhD thesis. Thank you.

Preamble:

Definition of Activity-based Learning: This is an established learning approach of actively engaging students in a classroom or outside the classroom to participate in the learning process through discussions, group or individual reflection, case studies, practical exercises, problem-based learning, etc.

PS: THIS MAY TAKE APPROXIMATELY 15 MINUTES TO COMPLETE

Part A – Demography of Students

1. Select Name of your University a. Ghana Technology Univ. Col. [ ] b. Valley View University [ ] c. Accra Institute of Technology [ ]
2. Program of Study a. Information Technology [ ] b. Computer Science [ ] c. Computer Engineering [ ] d. Information Systems [ ] e. Electrical Electronic Engineering [ ] f. Mobile Computing [ ]
3. Gender a. Male [ ] b. Female [ ]
4. Age of Participants a. 18- 20 [ ] b. 21-24 [ ] c. 25 – 30 [ ] d. 31- 35 [ ] e. 36 and above [ ]
5. Indicate your current level a. 100 [ ] b. 200 [ ] c. 300 [ ] d. 400 [ ]
6. Overall, how many courses are you enrolled for this semester? a. 1-2 [ ] b. 3-5 [ ] c. 6 or more [ ]
7. How many programming courses are you enrolled this semester? a. 1 [ ] b. 2 [ ] c. 3 [ ] d. 4 or above [ ]
8. Indicate the programming language(s) doing this semester
   a. C++ [ ] b. Java [ ] c. PhP [ ] d. C [ ] e. Python[ ] f. C# [ ] g. ASP.net [ ] h. Visual Basic.NET [ ] i. Other: ………………………………

9. What is your level of computer proficiency? a. Beginner [ ] b. Intermediate [ ] c. Advanced [ ]
10. Experience with internet a. less than 2 yrs. [ ] b. 2-3yrs [ ] c. 4-5yrs[ ] d. More than 5yrs [ ]

Part B: Experience of Activity-based Approach in Learning Programming

11. Which of the following best describes the activity-based approach currently used in your programming class? *Please tick only one where appropriate.*

Activity-based Approach Used	Never	Rarely	Occasionally	Frequently	Very Frequently
Case Study					
Quizzes					
Projects					
Group Discussions / Presentations					
Educational Games					
Videos					
Debates					
Problem Solving					
Round Table Discussions					
Peer Review					
Field Work					
Concept Mapping					
TEL Tools (Socrative, Padlet, Google Docs, etc.)					

12. Please indicate any other activity-based approach used in your class apart from the ones indicated above?

…………………………………………………………………………………………

…………………………………………………………………………………………

Part C: Challenges of Learning Programming

13. What kind of difficulty do you face in learning programming?

Statement	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Using program development environment					
Gaining access to computers/networks					
Understanding programming structures					
Learning the programming language syntax					
Designing a program to solve a certain task					
Dividing functionality into procedures					
Finding bugs from my own program					

14. Which programming concepts have been difficult for you to learn?

Topics / Concepts	Never	Seldom	Sometimes	Often	Almost Always
Variables (lifetime, scope)					
Selection structures					
Loop structures					
Recursion					
Arrays					
Function/Methods					
Pointers, references					
Parameters					
Structured data types					
Abstract data types					
Input/output handling					
Error handling					
Using language libraries					

Part D: General Comments

In this section, kindly provide your general perception or challenges on the activity-based approach currently used in your programming class.

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

Appendix C: Students' Interview Guide (First Semester)

Interview protocol used in semi-structured interviews/focused group discussions with students' participants.

- Name of HEI
- Gender
- Level
- Academic program enrolled

QUESTIONS (13 mins)

1. Tell me your learning experience in your programming class.
2. Do you have difficulties in learning programming? Why do you have difficulties with your programming assignments and activities if any?
3. Are you aware of the activity-based approach used by your lecturer in teaching your programming course? Can you give an example of the activity-based approach used by your programming lecturer(s)?
4. How do you get engaged in the activity-based approach in your class?
5. Does the BL environment support your activity-based learning experience?
6. What are the main challenges you and your peers struggle with in the activity-based approach employed in your class teaching programming?
7. Describe a situation you had difficulty with the activities in the blended learning environment and the steps you took to achieve your learning goals.

Appendix D: Faculty Interview Guide on the Curriculum Design (Semester 2)

Dear Colleague,

This interview aims at soliciting your views and experiences on a curriculum mediation for computer programming instruction towards a development of a ***Framework of Activity-Based Computer Programming Instructional Approach in a Blended Learning Environment of Higher Education Institutions in Ghana***. This research is intended for solely academic purposes, in view of that please indicate your candid knowledge and responses for the questions. Confidentiality and anonymity of your responses are assured and will be used only for writing my PhD thesis. Thank you.

1. Do you have the course outline for this course? Is the course outline aligned to the computer science curriculum in your institution?
2. Are you able to complete the learning outcomes of your course outline every semester?
3. Does the developed curriculum/syllabus/course outline support the activity-based approach in teaching programming in your university?
4. What are the current pedagogic approaches used to implement the programming class?
5. Have you identified any weakness in the current curriculum for teaching programming?
6. If you have the chance; which part of the curriculum will you change to enhance teaching and learning programming in HEIs? Which one will you recommend to be maintained?
7. How do you assess students using this curriculum? Why are you using that?
8. If you have the chance; which part of the assessment approach will you change to enhance hands-on experience on programming in HEIs? Which one will you recommend to be maintained?

General Comments. (5 mins)

Do you have any additional comment on how the current developed curriculum for teaching programming should be handled in HEIs?

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

Appendix E: Questionnaire for Respondents (Students) – After Second Semester

Dear Student,

Thank you for your corporation on the previous questionnaire you responded last semester. This questionnaire aims at soliciting your views and experiences towards a development of a ***Framework of Activity-Based Computer Programming Instructional Approach in a Blended Learning Environment of Higher Education Institutions in Ghana***. This research is intended for solely academic purposes, in view of that please indicate your candid knowledge and responses for the questions. Confidentiality and anonymity of your responses are assured and will be used only for writing my PhD thesis. Thank you.

PS: THIS MAY TAKE APPROXIMATELY 30 MINUTES TO COMPLETE

Instructions:

For each of the following statements from Part A to Part G, please tell me how true you think it is for you. The questions ask about your opinion. There are no right or wrong answers.

1- Strongly Disagree (SD) – if the statement does not apply to you at all.
2- Disagree (D) – if the statement occasionally applies to you.
3- Neutral (N) - if the statement neither applies to you nor applies.
4- Agree (A) – if the frequently applies to you
5- Strongly Agree (SA) – if the always applies to you.

Part A – Instructional Activities in Activity-based Blended Learning Environment

No.	Statements	SD	D	N	A	SA
1.	The instructor clearly communicated important course topics					
2.	The instructor clearly communicated important course goals.					
3.	The instructor clearly communicated important course topics					
4.	The instructor clearly communicated important due dates/time frames for learning activities.					
5.	The instructor was helpful in identifying areas of agreement and disagreement on course topics that helped me to learn.					

6.	The instructor was helpful in guiding the class towards understanding course topics in a way that helped me clarify my thinking.					
7.	The instructor helped to keep course participants engaged and participating in productive dialogue.					
8.	The instructor helped keep the course participants on task in a way that helped me to learn.					
9.	The instructor encouraged course participants to explore new concepts in this course.					
10.	Instructor actions reinforced the development of a sense of community among course participants.					
11.	The instructor helped to focus discussion on relevant issues in a way that helped me to learn.					
12.	The instructor provided feedback that helped me understand my strengths and weaknesses.					
13.	The instructor provided feedback in a timely fashion.					

Part B: Social Activities in Activity-based Blended Learning Environment

No.	**Statements**	SD	D	N	A	SA
14.	Getting to know other course participants gave me a sense of belonging in the course.					
15.	I am able to form distinct impressions of some course participants.					
16.	Online or web-based communication is an excellent medium for social interaction.					
17.	I felt comfortable conversing through the online medium.					
18.	I felt comfortable participating in the course discussions.					
19.	I felt comfortable interacting with other course participants					
20.	I felt comfortable disagreeing with other course participants while still maintaining a sense of trust.					
21.	I felt that my point of view was acknowledged by other course participants.					

22.	Online discussions help me to develop a sense of collaboration.					
23.	I felt comfortable to ask questions in the classroom					
24.	I felt comfortable to share my source codes to my colleagues					

Part C: Activity-based Approach Used

What is your view on the following statements on the activity-based approach used in your programming class?

No.	**Statements**	SD	D	N	A	SA
25.	The activity-based approach enables the student to learn at their own pace through teacher- facilitated engagement.					
26.	The different types of activities have helped me in retention of information					
27.	The activity-based approach has helped me in answering programming questions and case studies.					
28.	The activity-based approach is a waste of time.					
29.	The activity-based approach should be employed in all my courses					
30.	Activities given before the lecture are useful.					
31.	Activities given after the lecture are useful.					
32.	Both prior and after activities are useful					
33.	Had fun programming in class and after class because of the approach used by my instructor					
34.	Over all the activity-based approach used in class is good.					

Part D: Cognitive Activities in Activity-based Blended Learning Environment

No.	Statement	SD	D	N	A	SA
35.	The online and face-to-face activities is very engaging					
36.	Activities, Case studies, Class exercises and assignments posed increased my interest to programme.					
37.	Course activities irritated my curiosity to program both in the classroom and outside the class					

38.	I felt motivated to explore content related questions in programming					
39.	I utilized a variety of information sources to explore problems posed in this course.					
40.	Brainstorming and finding relevant information helped me resolve content related questions in the classroom and online					
41.	Online discussions were valuable in helping me appreciate different perspectives.					
42.	Combining new information helped me answer questions raised in course activities.					
43.	Learning activities helped me construct explanations/solutions. Reflection on course content and discussions helped me understand fundamental concepts in the programming class.					
44.	I can describe ways to test and apply the knowledge created in this course.					
45.	I have developed solutions to course problems that can be applied in practice.					
46.	I can apply the knowledge created in this course to my work or other non-class related activities.					

Part E: Learning Gains / Satisfaction

No.	Statement	SD	D	N	A	SA
47.	Overall, I am satisfied with this course					
48.	In your opinion do you think the activity-based approach in teaching programming is effective.					
49.	I am satisfied with how I am fully engaged both online and face-to-face					
50.	I can see improvement in my programming skills					
51.	I am comfortable with the approach used in teaching programming.					
52.	The online platform is stable					
53.	The online platform is user friendly					

54.	The learning environment in the classroom promotes activity learning					
55.	I am satisfied with the time and credit hours allocated for this course					
56.	I am comfortable with due dates and time on the online platform					
57.	I am satisfied with the instructional approach					
58.	I am satisfied with the support I again from my university on my learning experience.					

Part F: Critical Skills Development, Learning and Knowledge Attainment

No.	Statements	SD	D	N	A	SA
59.	I learned much in this course.					
60.	I felt comfortable disagreeing with other course participants while still maintaining a sense of trust.					
61.	I had fun programming in class while my lecturer guides me					
62.	I am comfortable with group activities on a case study or a question.					
63.	I can program with least or no supervision after class and online engagement					
64.	Felt like I made a meaningful contribution to the activities in class					
65.	Felt challenged by the approach used in teaching the programming course					
66.	Felt interested in the material					
67.	Didn’t see its immediate relevance to my work as a student					
68.	Didn’t see its relevance to my future career					

Part G: Feedback Activities on the Blended Learning Environment

No.	Statements	SD	D	N	A	SA
69.	Found that group work is dominated by more vocal people which intimidates or put me off to contribute or grant feedback to my colleagues and instructor.					

70.	Feedback from my instructor is very effective					
71.	Believed that the lecturer(s) were motivated by the activities					
72.	My performance was directly related to the feedbacks that the activity-based learning provided					
73.	Was provided with all the materials/texts that I needed and was communicated promptly					
74.	Assessments, assignments, case studies, projects are well communicated and results or feedback issued on time					

Part H: General Commitments

In this section, kindly provide your general perception on the activity-based approach employed in your programming class.

…………………………………………………………………………………………………

…………………………………………………………………………………………………

Appendix F: Students Interview Guide on the ABLA Framework (Semester 3)

Interview protocol used in semi-structured interviews/focused group discussions with students' participants.

- Name of HEI
- Gender
- Level
- Degree Program

Questions on the evaluation of the ABLA framework in BL for Programming Instruction (10 min)

1. Do your instructor use activity-based approach in the teaching and learning of computer programming? Why do you say so?
2. What activity-based approach do you use? How often?
3. What are the some of the learning gains you benefit from the instructional activities used in your class? Why do you say so? (why do you not?)
4. What are the some of the learning gains you benefit from the social activities you use in your class? Why do you use that? (why do you not?)
5. What are the some of the learning gains you benefit from the cognitive activities you use in your class? Why do you use that? (why do you not?)
6. Tell me, how does your instructors assess you in your class? How often do you get feedback(s) from your lecturers?
7. Do you have learner support unit at your university? How helpful are they or otherwise, indicate to me?
8. How are your students benefiting from the activity-based approach used in your class?
9. What have been the challenges from using the activity-based approach?

General Comments. (5 mins)

Do you have any additional comment on how the ABLA Framework for teaching programming should be handled in HEIs? …………………………………………

……………………………………………………………………………………………

Appendix G: Faculty Interview Guide on the ABLA Framework (Semester 3)

Interview protocol used in semi-structured interviews with faculty participants.

- Gender
- Institution
- Number of years teaching in HEI
- Number of years on blended learning
- Programming course(s) taught

Context of the Activity-Based Approach in BL Environment (8 mins)

1. Can you tell me about your experience with the activity-based approach within the BL environment? Context (5 min)
2. What is your course about?
3. About how many students do you have in each section?
4. Where is the course situated in their programs? (required, optional, mostly freshman, mostly seniors, etc?)
5. How long have you been teaching this course?

Evaluation of the ABLA framework for Programming Instruction (10 min)

1. Do you use activity-based approach in teaching programming? Why do you use that (why do you not?)
2. Are you still using the curriculum / course outline developed for teaching your class?
3. What activity-based approach do you use? How often? Why do you choose to use that approach?
4. What are the some of the instructional activities you use in your class? Why do you use that? (why do you not?)
5. What are the some of the social activities you use in your class? Why do you use that? (why do you not?)
6. What are the some of the cognitive activities you use in your class? Why do you use that? (why do you not?)
7. Tell me, how do you usually perform assessment? How often do you grant feedback to

your students?

8. Do you have learner support unit at your university? How helpful are they or otherwise, indicate to me?

Learning Gains (10 mins)

1. What do you think about your activity-based approach? Has it worked well? Why or why not?
2. What are the teaching and learning gains for using the activity-based approach in teaching programming?
3. How are your students benefiting from the activity-based approach used in your class?
4. What have been the challenges from using the activity-based approach?
5. What would you like it to be able to do if it could be possible?

Efficiency (10 mins)

1. How effective is the activity-based framework used in teaching programming?
2. Does your institutions LMS help you to be more efficient within the BL context? How?
3. Which features save you the most time for the engagement you afford the students?

Learning (20 mins)

1. Do you feel that using the activity-based approach helps you teach more effectively? In what ways? How do you use it?
2. Do you feel that using the activity-based approach helps students learn more effectively? In what ways?
3. How do you engage the students within the face-to-face instruction and the online platform?
4. How do you think a BL environment like your institutions could be used to improve learning in your subject area?
5. If you use the discussion/chat tools, how do you use them? What has worked well? What hasn't worked well?
6. Is your decision to adopt activity-based approach within the BL environment influenced

by any internal or external factors and what are they?

General Comments. (5 mins)

Do you have any additional comment on how the activity-based approach for teaching programming should be handled in HEIs?

……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………

Appendix H: Permission Letters

UNISA university of south africa

21st May, 2019

The Head of Department
Department of Computer Science
Ghana Technology University College
PMB 100, Accra. (kquist-aphetsi@gtuc.edu.gh)

Dear Dr. Kester,

REQUEST FOR PERMISSION TO CONDUCT RESEARCH AT GHANA TECHNOLOGY UNIVERSITY COLLEGE, GTUC (DEPARTMENT OF COMPUTER SCIENCE)

I, Emmanuel Freeman is doing a research with Prof. Ian Sanders a Professor and Dr. Bester Chimbo the Chair of School of Computing (SOC) Ethical Committee, in the Department of Information Systems towards a PhD at the University of South Africa on the topic: **A Framework for an Activity-Based Computer Programming Instruction in a Blended Learning Environment of Higher Education Institutions in Ghana.**

The aim of the study is to develop a framework for an activity-based computer programming instruction in a blended learning environment of HEIs in Ghana. Ghana Technology University College has been selected because it is seen as one of the best technology oriented intuitions in Ghana that employs blended learning in Ghana.

The study will entail faculty and students teaching and learning experience within the blended learning environment. The benefits of this study are to establish a robust framework that can be used in teaching computer programming courses in HEIs using activity-based learning approach in a blended learning environment, develop a pedagogic approach for instructing programming and also establish learning gains on the students and lecturers for using activity based approach in teaching programming.

The study will not pose any threat to any respondent since data receive from the students and faculty shall be strictly confidential and anonymous.

Yours sincerely,

....
Emmanuel Freeman
PhD. Candidate, School of Computing, UNISA

HEAD, OF COMPUTER SCIENCE DEPT.
GHANA TECHNOLOGY UNIV. COLLEGE
P. M. B. 100, ACCRA

University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane

UNISA university of south africa

10th August, 2019.

The Head of Department
Department of Computer Science and Information Technology
Valley View University, Oyibi-Accra
papaprince@vvu.edu.gh

Dear Sir,

REQUEST FOR PERMISSION TO CONDUCT RESEARCH AT VALLEY VIEW UNIVERSITY (DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY).

I, Emmanuel Freeman is doing a research with Prof. Ian Sanders a Professor and Dr. Bester Chimbo the Chair of School of Computing (SOC) Ethical Committee, in the Department of Information Systems towards a PhD at the University of South Africa on the topic: **A Framework for Activity-Based Computer Programming Instruction in a Blended Learning Environment of Higher Education Institutions in Ghana.**

The aim of the study is to develop a framework for an activity-based computer programming instruction in a blended learning environment of HEIs in Ghana. Valley View University has been selected because it is seen as one of the best technology oriented intuitions in Ghana that employs blended learning in Ghana.

The study will entail faculty and students teaching and learning experience within the blended learning environment. The benefits of this study are to establish a robust framework that can be used in teaching computer programming courses in HEIs using activity-based learning approach in a blended learning environment, develop a pedagogic approach for instructing programming and also establish learning gains on the students and lecturers for using activity based approach in teaching programming.

The study will not pose any threat to any respondent since data receive from the students and faculty shall be strictly confidential and anonymous.

Yours sincerely,

....
Emmanuel Freeman
PhD. Candidate, School of Computing, UNISA

Prince Y. O. Amoako
Head, IT Department VVU

University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa
Telephone: +27 12 429 3111 Facsimile: +27 12 429 4150
www.unisa.ac.za

21st May, 2019

The Registrar
Accra Institute of Technology (AIT)
P.O. Box AN-19782
Accra-North
osei-boakye@ait.edu.gh / +233 577668801

Dear Mr. Dominic Osei-Boakye,

REQUEST FOR PERMISSION TO CONDUCT RESEARCH AT ACCRA INSTITUTE OF TECHNOLOGY (AIT).

I, Emmanuel Freeman is doing a research with Prof. Ian Sanders a Professor and Dr. Bester Chimbo the Chair of School of Computing (SOC) Ethical Committee, in the Department of Information Systems towards a PhD at the University of South Africa on the topic: **A Framework for an Activity-Based Computer Programming Instruction in a Blended Learning Environment of Higher Education Institutions in Ghana.**

The aim of the study is to develop a framework for an activity-based computer programming instruction in a blended learning environment of HEIs in Ghana. Accra Institute of Technology has been selected because it is seen as one of the best technology oriented intuitions in Ghana that employs blended learning in Ghana.

The study will entail faculty and students teaching and learning experience within the blended learning environment. The benefits of this study are to establish a robust framework that can be used in teaching computer programming courses in HEIs using activity-based learning approach in a blended learning environment, develop a pedagogic approach for instructing programming and also establish learning gains on the students and lecturers for using activity based approach in teaching programming.

The study will not pose any threat to any respondent since data receive from the students and faculty shall be strictly confidential and anonymous.

Yours sincerely,

Emmanuel Freeman
PhD. Candidate, School of Computing, UNISA

Dominic Osei-Boakye
Registrar (AIT)

University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa
Telephone: +27 12 429 3111 Facsimile: +27 12 429 4150
www.unisa.ac.za

Appendix I: Communalities of the SEM of the Data

Communalities		
	Initial	**Extraction**
Cognitive 6	1.000	.717
Cognitive 8	1.000	.818
Cognitive 10	1.000	.711
Cognitive 13	1.000	.761
Activity-based Approach Used 5	1.000	.799
Activity-based Approach Used 6	1.000	.696
Activity-based Approach Used 12	1.000	.764
Satisfaction 1	1.000	.749
Satisfaction 5	1.000	.737
Satisfaction 6	1.000	.686
Satisfaction 8	1.000	.776
Satisfaction 11	1.000	.719
Satisfaction 12	1.000	.768
Instructional Activities 5	1.000	.810
Instructional Activities 6	1.000	.872
Instructional Activities 8	1.000	.838
Instructional Activities 10	1.000	.838
Social 2	1.000	.723
Social 4	1.000	.787
Social 5	1.000	.721
Feedback 2	1.000	.829
Feedback 3	1.000	.760
Feedback 7	1.000	.738
Skills Development 6	1.000	.791
Skills Development 7	1.000	.772
Skills Development 8	1.000	.764
Extraction Method: Principal Component Analysis.		

Appendix J: Assessment of Normality

Variable Code	Variables	Min	Max	Skew	C.R.	Kurtosis	C.R.
AU12	Brainstorming and finding relevant information helped me resolve content related questions in the classroom and online	1.000	5.000	.873	6.176	-.170	-.601
AU5	Combining new information helped me answer questions raised in course activities	1.000	5.000	.924	6.535	-.061	-.215
AU6	Reflection on course content and discussions helped me understand fundamental concepts in the computer programming class	1.000	5.000	.222	1.573	-1.135	-4.014
CA10	I can apply the knowledge created in this course to my current work or other non-class related activities	1.000	5.000	-.648	-4.584	.056	.199
CA13	Educational Games	1.000	5.000	-.576	-4.071	-.309	-1.093
CA6	Concept Mapping	1.000	5.000	-.692	-4.890	.554	1.960
CA8	Videos	1.000	5.000	-.545	-3.854	-.081	-.285
FB2	Overall, I am satisfied with this course	1.000	5.000	-.643	-4.545	.231	.816
FB3	I am comfortable with the approach used in teaching computer progra mming	1.000	5.000	-.524	-3.702	.375	1.326
FB7	The online platform is stable	1.000	5.000	-.584	-4.130	.145	.511
IA10	The learning environment in the classroom promotes activity- based learning	1.000	5.000	-.427	-3.020	-.415	-1.468
IA5	I am satisfied with the instructional approach	1.000	5.000	-.609	-4.307	-.022	-.077

Variable Code	Variables	Min	Max	Skew	C.R.	Kurtosis	C.R.
IA6	I am satisfied with the support I gain from my university on my learning experience	1.000	5.000	-.494	-3.495	-.318	-1.124
IA8	The instructor was helpful in guiding the class towards understanding course topics in a way that helped me clarify my thinking	1.000	5.000	-.584	-4.128	-.280	-.989
LG1	The instructor encouraged course participants to explore new concepts in this course	1.000	5.000	-.387	-2.733	-.314	-1.112
LG11	The instructor helped to focus discussion on relevant issues in a way that helped me to learn	1.000	5.000	-.533	-3.770	-.010	-.036
LG12	The instructor helped to keep course participants engaged and participating in productive dialogue	1.000	5.000	-.613	-4.331	-.092	-.326
LG5	I am able to form distinct impressions of some course participants	1.000	5.000	-.424	-2.995	-.236	-.835
LG6	I felt comfortable conversing through the online medium	1.000	5.000	-.499	-3.526	-.362	-1.281
LG8	I felt comfortable participating in the course discussions.	1.000	5.000	-.575	-4.069	.248	.876
SA2	Feedback from my instructor is very effective The feedback from my peers was very effective	1.000	5.000	-.600	-4.240	.245	.865
SA4	Feedback from my instructor is very effective	1.000	5.000	-.764	-5.405	.702	2.480

Variable Code	Variables	Min	Max	Skew	C.R.	Kurtosis	C.R.
	The feedback from my peers was very effective						
SA5	Assessments, assignments, case studies, projects are well communicated and results or feedback issued on time	1.000	5.000	-.927	-6.552	1.182	4.181
SD6	I felt like I made a meaningful contribution to the activities in class	1.000	5.000	-.577	-4.078	.214	.755
SD7	I felt challenged by the approach used in teaching the programming course	1.000	5.000	-.657	-4.645	.040	.142
SD8	I felt like I made a meaningful contribution to the activities in class	1.000	5.000	-.492	-3.481	-.023	-.080
Multivariate						**272.824**	**61.920**

Abbreviations: AU = Activity Based Learning Used, IA = Instructional Activities, SA = Social Activities, LG = Learning Gains, SD = Skills Development, FB = Feedback, CA = Cognitive Activities

Appendix K: Sample Codes for the ABPI Platform

1. User Registration

```
<?php
$comp_model = new SharedController;
$page_element_id = "add-page-" . random_str();
$current_page = $this->set_current_page_link();
$csrf_token = Csrf::$token;
$show_header = $this->show_header;
$view_title = $this->view_title;
$redirect_to = $this->redirect_to;
?>
<section class="page" id="<?php echo $page_element_id; ?>" data-page-type="add" data-display-type="" data-page-url="<?php print_link($current_page); ?>">
    <?php
    if( $show_header == true ){
    ?>
    <div  class="bg-light p-3 mb-3">
        <div class="container">
            <div class="row ">
                <div class="col ">
                    <h4 class="record-title">User registration</h4>
                </div>
                <div class="col-sm-6 comp-grid">
                    <div class="">
                        <div class="text-center">
                            Already have an account?  <a class="btn btn-primary" href="<?php print_link('') ?>"> Login</a>
                        </div>
                    </div>
                </div>
            </div>
        </div>
```

```
</div>
<?php
}
?>
<div  class="">
    <div class="container">
        <div class="row ">
            <div class="col-md-7 comp-grid">
                <?php $this :: display_page_errors(); ?>
                <div  class="bg-light p-3 animated fadeIn page-content">
                    <form id="admin-userregister-form" role="form" novalidate enctype="multipart/form-data" class="form page-form form-horizontal needs-validation" action="<?php print_link("index/register?csrf_token=$csrf_token") ?>" method="post">
                        <!--[main-form-start]-->
                        <div>
                            <div class="form-group ">
                                <div class="row">
                                    <div class="col-sm-4">
                                        <label class="control-label" for="user_name">User Name </label>
                                    </div>
                                    <div class="col-sm-8">
                                        <div class="">
                                            <input id="ctrl-user_name" value="<?php echo $this->set_field_value('user_name',""); ?>" type="text" placeholder="Enter User Name" name="user_name" data-url="api/json/admin_user_name_value_exist/" data-loading-msg="Checking availability ..." data-available-msg="Available" data-unavailable-msg="Not available" class="form-control  ctrl-check-duplicate" />
                                                <div class="check-status"></div>
                                            </div>
                                        </div>
                                    </div>
                                </div>
                                <div class="form-group ">
```

```
<div class="row">
    <div class="col-sm-4">
        <label class="control-label" for="email">Email </label>
    </div>
    <div class="col-sm-8">
        <div class="">
            <input id="ctrl-email" value="<?php echo $this->set_field_value('email',""); ?>" type="email" placeholder="Enter Email" name="email" data-url="api/json/admin_email_value_exist/" data-loading-msg="Checking availability ..." data-available-msg="Available" data-unavailable-msg="Not available" class="form-control ctrl-check-duplicate" />
                <div class="check-status"></div>
            </div>
        </div>
    </div>
</div>
<div class="form-group ">
    <div class="row">
        <div class="col-sm-4">
            <label class="control-label" for="password">Password </label>
        </div>
        <div class="col-sm-8">
            <div class="input-group">
                <input id="ctrl-password" value="<?php echo $this->set_field_value('password',""); ?>" type="password" placeholder="Enter Password" name="password" class="form-control password password-strength" />
                    <div class="input-group-append cursor-pointer btn-toggle-password">
                        <span class="input-group-text"><i class="fa fa-eye"></i></span>
                    </div>
                </div>
                <div class="password-strength-msg">
```

```
                                        <small                    class="font-weight-bold">Should
contain</small>
                                        <small       class="length       chip">6       Characters
minimum</small>
                                        <small class="caps chip">Capital Letter</small>
                                        <small class="number chip">Number</small>
                                        <small class="special chip">Symbol</small>
                                    </div>
                                </div>
                            </div>
                        </div>
                        <div class="form-group ">
                            <div class="row">
                                <div class="col-sm-4">
                                    <label                                  class="control-label"
for="confirm_password">Confirm Password </label>
                                </div>
                                <div class="col-sm-8">
                                    <div class="input-group">
                                        <input  id="ctrl-password-confirm"  data-match="#ctrl-
password"          class="form-control       password-confirm       "       type="password"
name="confirm_password"  placeholder="Confirm Password" />
                                        <div   class="input-group-append   cursor-pointer   btn-
toggle-password">
                                            <span    class="input-group-text"><i    class="fa    fa-
eye"></i></span>
                                        </div>
                                        <div class="invalid-feedback">
                                            Password does not match
                                        </div>
                                    </div>
                                </div>
                            </div>
                        </div>
```

```
<div class="form-group ">
    <div class="row">
        <div class="col-sm-4">
            <label class="control-label" for="photo">Photo </label>
        </div>
        <div class="col-sm-8">
            <div class="">
                <div class="dropzone " input="#ctrl-photo" fieldname="photo" data-multiple="false" dropmsg="Choose files or drag and drop files to upload" btntext="Browse" extensions=".jpg,.png,.gif,.jpeg" filesize="3" maximum="1">
                    <input name="photo" id="ctrl-photo" class="dropzone-input form-control" value="<?php echo $this->set_field_value('photo',""); ?>" type="text" />
                        <!--<div class="invalid-feedback animated bounceIn text-center">Please a choose file</div>-->
                        <div class="dz-file-limit animated bounceIn text-center text-danger"></div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
<!--[main-form-end]-->
<div class="form-group form-submit-btn-holder text-center mt-3">
    <button class="btn btn-primary" type="submit">
        Submit
        <i class="fa fa-send"></i>
    </button>
</div>
</form>
</div>
</div>
```

```
                </div>
              </div>
            </div>
          </section>
```

2. Add Activity Sample Codes

```
<?php
$comp_model = new SharedController;
$page_element_id = "add-page-" . random_str();
$current_page = $this->set_current_page_link();
$csrf_token = Csrf::$token;
$show_header = $this->show_header;
$view_title = $this->view_title;
$redirect_to = $this->redirect_to;
?>
<section class="page" id="<?php echo $page_element_id; ?>" data-page-type="add"  data-display-type="" data-page-url="<?php print_link($current_page); ?>">
  <?php
  if( $show_header == true ){
  ?>
  <div  class="bg-light p-3 mb-3">
    <div class="container">
      <div class="row ">
        <div class="col ">
          <h4 class="record-title">Add New Activities</h4>
        </div>
      </div>
    </div>
  </div>
  <?php
  }
  ?>
  <div  class="">
```

```
<div class="container">
    <div class="row ">
        <div class="col-md-7 comp-grid">
            <?php $this :: display_page_errors(); ?>
            <div  class="bg-light p-3 animated fadeIn page-content">
                <form id="activities-add-form" role="form" novalidate enctype="multipart/form-data" class="form page-form form-horizontal needs-validation" action="<?php print_link("activities/add?csrf_token=$csrf_token") ?>" method="post">
                    <div>
                        <div class="form-group ">
                            <div class="row">
                                <div class="col-sm-4">
                                    <label class="control-label" for="owner_id">Owner Id </label>
                                </div>
                                <div class="col-sm-8">
                                    <div class="">
                                        <input id="ctrl-owner_id" value="<?php echo $this->set_field_value('owner_id',""); ?>" type="number" placeholder="Enter Owner Id" step="1" name="owner_id"  class="form-control " />
                                    </div>
                                </div>
                            </div>
                        </div>
                        <div class="form-group ">
                            <div class="row">
                                <div class="col-sm-4">
                                    <label class="control-label" for="topic_id">Topic Id </label>
                                </div>
                                <div class="col-sm-8">
                                    <div class="">
                                        <input id="ctrl-topic_id" value="<?php echo $this->set_field_value('topic_id',""); ?>" type="number" placeholder="Enter Topic Id" step="1" name="topic_id"  class="form-control " />
                                    </div>
```

```
                    </div>
                  </div>
                </div>
                <div class="form-group ">
                  <div class="row">
                    <div class="col-sm-4">
                      <label class="control-label" for="title">Title </label>
                    </div>
                    <div class="col-sm-8">
                      <div class="">
                        <input id="ctrl-title" value="<?php echo $this-
>set_field_value('title',""); ?>" type="text" placeholder="Enter Title" name="title"
class="form-control " />
                        </div>
                      </div>
                    </div>
                  </div>
                  <div class="form-group ">
                    <div class="row">
                      <div class="col-sm-4">
                        <label class="control-label" for="description">Description
</label>
                      </div>
                      <div class="col-sm-8">
                        <div class="">
                          <textarea placeholder="Enter Description" id="ctrl-
description" rows="5" name="description" class=" form-control"><?php echo $this-
>set_field_value('description',""); ?></textarea>
                          <!--<div class="invalid-feedback animated bounceIn
text-center">Please enter text</div>-->
                        </div>
                      </div>
                    </div>
                  </div>
```

```
<div class="form-group ">
    <div class="row">
        <div class="col-sm-4">
            <label class="control-label" for="files">Files </label>
        </div>
        <div class="col-sm-8">
            <div class="">
                <textarea placeholder="Enter Files" id="ctrl-files" rows="5" name="files" class=" form-control"><?php  echo $this->set_field_value('files',""); ?></textarea>
                <!--<div class="invalid-feedback animated bounceIn text-center">Please enter text</div>-->
            </div>
        </div>
    </div>
</div>
<div class="form-group ">
    <div class="row">
        <div class="col-sm-4">
            <label class="control-label" for="date_created">Date Created </label>
        </div>
        <div class="col-sm-8">
            <div class="input-group">
                <input id="ctrl-date_created" class="form-control datepicker  datepicker" value="<?php  echo $this->set_field_value('date_created',""); ?>" type="datetime"  name="date_created" placeholder="Enter Date Created" data-enable-time="true" data-min-date="" data-max-date="" data-date-format="Y-m-d H:i:S" data-alt-format="F j, Y - H:i" data-inline="false" data-no-calendar="false" data-mode="single" />
                    <div class="input-group-append">
                        <span class="input-group-text"><i class="fa fa-calendar"></i></span>
                    </div>
                </div>
```

```
                                        </div>
                                    </div>
                                </div>
                                <div class="form-group ">
                                    <div class="row">
                                        <div class="col-sm-4">
                                            <label class="control-label" for="course_id">Course Id
</label>
                                        </div>
                                        <div class="col-sm-8">
                                            <div class="">
                                                <input id="ctrl-course_id"  value="<?php  echo $this-
>set_field_value('course_id',""); ?>" type="number" placeholder="Enter Course Id" step="1"
name="course_id"  class="form-control " />
                                                </div>
                                            </div>
                                        </div>
                                    </div>
                                </div>
                                <div class="form-group form-submit-btn-holder text-center mt-
3">
                                    <div class="form-ajax-status"></div>
                                    <button class="btn btn-primary" type="submit">
                                        Submit
                                        <i class="fa fa-send"></i>
                                    </button>
                                </div>
                            </form>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </section>
```

3. Teacher's Portal

```
<?php
/**
 * Topics Page Controller
 * @category  Controller
 */
class TopicsController extends SecureController{
	function __construct(){
		parent::__construct();
		$this->tablename = "topics";
	}
	/**
   * List page records
   * @param $fieldname (filter record by a field)
   * @param $fieldvalue (filter field value)
   * @return BaseView
   */
	function index($fieldname = null , $fieldvalue = null){
		$request = $this->request;
		$db = $this->GetModel();
		$tablename = $this->tablename;
		$fields = array("topics.topic_id",
			"topics.owner_id",
			"topics.title",
			"topics.description",
			"teachers.user_name AS teachers_user_name",
			"topics.status",
			"topics.date_created");
		$pagination = $this->get_pagination(MAX_RECORD_COUNT); // get
current pagination e.g array(page_number, page_limit)
		//search table record
		if(!empty($request->search)){
			$text = trim($request->search);
```

```
$search_condition = "(
        topics.topic_id LIKE ? OR
        topics.owner_id LIKE ? OR
        topics.course_id LIKE ? OR
        topics.title LIKE ? OR
        topics.description LIKE ? OR
        teachers.user_name LIKE ? OR
        topics.status LIKE ? OR
        topics.date_created LIKE ? OR
        teachers.teacher_id LIKE ? OR
        teachers.course_id LIKE ? OR
        teachers.email LIKE ? OR
        teachers.password LIKE ? OR
        teachers.photo LIKE ? OR
        teachers.date_added LIKE ? OR
        teachers.status LIKE ?
    )";
    $search_params = array(

"%$text%","%$text%","%$text%","%$text%","%$text%","%$text%","%$tex
t%","%$text%","%$text%","%$text%","%$text%","%$text%","%$text%","%$text%
","%$text%"
    );
    //setting search conditions
    $db->where($search_condition, $search_params);
     //template to use when ajax search
    $this->view->search_template = "topics/search.php";
}
$db->join("teachers",     "topics.owner_id     =     teachers.teacher_id",
"INNER");
if(!empty($request->orderby)){
    $orderby = $request->orderby;
    $ordertype    =    (!empty($request->ordertype)    ?    $request-
>ordertype : ORDER_TYPE);
```

```
			$db->orderBy($orderby, $ordertype);
		}
		else{
			$db->orderBy("topics.topic_id", ORDER_TYPE);
		}
		if($fieldname){
			$db->where($fieldname , $fieldvalue); //filter by a single field
name
		}
		$tc = $db->withTotalCount();
		$records = $db->get($tablename, $pagination, $fields);
		$records_count = count($records);
		$total_records = intval($tc->totalCount);
		$page_limit = $pagination[1];
		$total_pages = ceil($total_records / $page_limit);
		$data = new stdClass;
		$data->records = $records;
		$data->record_count = $records_count;
		$data->total_records = $total_records;
		$data->total_page = $total_pages;
		if($db->getLastError()){
			$this->set_page_error();
		}
		$page_title = $this->view->page_title = "Topics";
		$this->render_view("topics/list.php", $data); //render the full page
	}
	/**
   * View record detail
	 * @param $rec_id (select record by table primary key)
   * @param $value value (select record by value of field name(rec_id))
   * @return BaseView
   */
	function view($rec_id = null, $value = null){
		$request = $this->request;
```

```
$db = $this->GetModel();
$rec_id = $this->rec_id = urldecode($rec_id);
$tablename = $this->tablename;
$fields = array("topics.topic_id",
        "topics.owner_id",
        "topics.course_id",
        "topics.title",
        "topics.description",
        "topics.status",
        "topics.date_created",
        "teachers.teacher_id AS teachers_teacher_id",
        "teachers.course_id AS teachers_course_id",
        "teachers.user_name AS teachers_user_name",
        "teachers.email AS teachers_email",
        "teachers.password AS teachers_password",
        "teachers.photo AS teachers_photo",
        "teachers.date_added AS teachers_date_added",
        "teachers.status AS teachers_status");
if($value){
        $db->where($rec_id, urldecode($value)); //select record based
on field name
}
else{
        $db->where("topics.topic_id", $rec_id);; //select record based
on primary key
}
$db->join("teachers", "topics.owner_id = teachers.teacher_id", "INNER
");
$record = $db->getOne($tablename, $fields );
if($record){
        $page_title = $this->view->page_title = "View  Topics";
}
else{
        if($db->getLastError()){
```

```
					$this->set_page_error();
				}
				else{
					$this->set_page_error("No record found");
				}
			}
			return $this->render_view("topics/view.php", $record);
		}
		/**
	 * Insert new record to the database table
		 * @param $formdata array() from $_POST
	 * @return BaseView
	 */
		function add($formdata = null){
			if($formdata){
				$db = $this->GetModel();
				$tablename = $this->tablename;
				$request = $this->request;
				//fillable fields
				$fields = $this->fields = array("owner_id","course_id","title","description","status","date_created");
				$postdata = $this->format_request_data($formdata);
				$this->rules_array = array(
					'owner_id' => 'numeric',
					'course_id' => 'numeric',
				);
				$this->sanitize_array = array(
					'owner_id' => 'sanitize_string',
					'course_id' => 'sanitize_string',
					'title' => 'sanitize_string',
					'description' => 'sanitize_string',
					'status' => 'sanitize_string',
					'date_created' => 'sanitize_string',
				);
```

```
			$this->filter_vals = true; //set whether to remove empty fields
			$modeldata = $this->modeldata = $this->validate_form($postdata);
			if($this->validated()){
				$rec_id = $this->rec_id = $db->insert($tablename, $modeldata);
				if($rec_id){
					$this->set_flash_msg("Record added successfully", "success");
					return  $this->redirect("topics");
				}
				else{
					$this->set_page_error();
				}
			}
		}
		$page_title = $this->view->page_title = "Add New Topics";
		$this->render_view("topics/add.php");
	}
	/**
  * Update table record with formdata
	 * @param $rec_id (select record by table primary key)
	 * @param $formdata array() from $_POST
  * @return array
  */
	function edit($rec_id = null, $formdata = null){
		$request = $this->request;
		$db = $this->GetModel();
		$this->rec_id = $rec_id;
		$tablename = $this->tablename;
		 //editable fields
		$fields = $this->fields = array("topic_id","owner_id","course_id","title","description","status","date_created");
		if($formdata){
```

```
$postdata = $this->format_request_data($formdata);
$this->rules_array = array(
	'owner_id' => 'numeric',
	'course_id' => 'numeric',
);
$this->sanitize_array = array(
	'owner_id' => 'sanitize_string',
	'course_id' => 'sanitize_string',
	'title' => 'sanitize_string',
	'description' => 'sanitize_string',
	'status' => 'sanitize_string',
	'date_created' => 'sanitize_string',
);
$modeldata = $this->modeldata = $this-
>validate_form($postdata);
if($this->validated()){
	$db->where("topics.topic_id", $rec_id);;
	$bool = $db->update($tablename, $modeldata);
	$numRows = $db->getRowCount(); //number of
affected rows. 0 = no record field updated
	if($bool && $numRows){
		$this->set_flash_msg("Record updated
successfully", "success");
		return $this->redirect("topics");
	}
	else{
		if($db->getLastError()){
			$this->set_page_error();
		}
		elseif(!$numRows){
			//not an error, but no record was updated
			$page_error = "No record updated";
			$this->set_page_error($page_error);
```

```
                                                $this->set_flash_msg($page_error,
"warning");
                                                return  $this->redirect("topics");
                                        }
                                }
                        }
                }
                $db->where("topics.topic_id", $rec_id);;
                $data = $db->getOne($tablename, $fields);
                $page_title = $this->view->page_title = "Edit  Topics";
                if(!$data){
                        $this->set_page_error();
                }
                return $this->render_view("topics/edit.php", $data);
        }
        /**
    * Delete record from the database
         * Support multi delete by separating record id by comma.
    * @return BaseView
    */
        function delete($rec_id = null){
                Csrf::cross_check();
                $request = $this->request;
                $db = $this->GetModel();
                $tablename = $this->tablename;
                $this->rec_id = $rec_id;
                //form multiple delete, split record id separated by comma into array
                $arr_rec_id = array_map('trim', explode(",", $rec_id));
                $db->where("topics.topic_id", $arr_rec_id, "in");
                $bool = $db->delete($tablename);
                if($bool){
                        $this->set_flash_msg("Record         deleted         successfully",
"success");
                }
```

```
                elseif($db->getLastError()){
                        $page_error = $db->getLastError();
                        $this->set_flash_msg($page_error, "danger");
                }
                return  $this->redirect("topics");
        }
}
```

Appendix L: Poster Presentation at SAICSIT Conference (2019), South Africa