

**AKENTEN APPIAH-MENKA UNIVERSITY
OF SKILLS TRAINING AND ENTREPRENEURIAL DEVELOPMENT**

**EXAMINING THE EFFECT OF SELF-CONCEPT, STUDENTS' COMPUTER
PROGRAMMING INTEREST AND COMPUTER PROGRAMMING ANXIETY
ON STUDENTS' COMPUTER PROGRAMMING ACHIEVEMENT**

MICHEAL APPIAH

MASTER OF SCIENCE

2023

**AKENTEN APPIAH-MENKA UNIVERSITY
OF SKILLS TRAINING AND ENTREPRENEURIAL DEVELOPMENT**

**EXAMINING THE EFFECT OF SELF-CONCEPT, STUDENTS' COMPUTER
PROGRAMMING INTEREST AND COMPUTER PROGRAMMING ANXIETY
ON STUDENTS' COMPUTER PROGRAMMING ACHIEVEMENT**

MICHEAL APPIAH

**A thesis in the Department of Information Technology Education, Faculty of
Applied Sciences and Mathematics Education, submitted to the School of
Graduate Studies in partial fulfilment
of the requirements for the award of the degree of
Master of Science
(Information Technology Education)
in the Akenten Appiah-Menka University of Skills Training and Entrepreneurial
Development**

2023

DECLARATION

STUDENT'S DECLARATION

I, Michael Appiah declare that this thesis, with the exception of quotations and references contained in published works which have all been identified and duly acknowledged, is entirely my own original work, and it has not been submitted, either in part or whole, for another degree elsewhere.

Signature..... Date: **06-10-2023**

SUPERVISOR'S DECLARATION

We hereby declare that the preparation and presentation of this work was supervised in accordance with the guidelines for supervision of thesis/dissertation/project as laid down by the Akenten Appiah-Menka University of Skills Training and Entrepreneurial Development.

Signature..... Date: **06-10-2023**

Professor Francis Ohene Boateng

ACKNOWLEDGMENT

I extend my deepest appreciation to all those whose support and contributions have played a pivotal role in the realization of this project. To my supervisor, whose guidance and wisdom have been a constant source of inspiration, shaping my understanding and enhancing the quality of my work. I am profoundly grateful for the unwavering support of my friends and family, whose encouragement and belief in my abilities have been a driving force throughout this journey.

DEDICATION

In profound gratitude and heartfelt appreciation, I dedicate this work to those whose unwavering support and encouragement have fueled the journey of my aspirations. To my family, whose love is my anchor and source of strength, and whose sacrifices have paved the way for my pursuits. To my supervisor, whose wisdom has illuminated my path and guided me through challenges with invaluable insights. To my friends, whose camaraderie has added joy to the pursuit of my dreams.

TABLE OF CONTENTS

DECLARATION	iii
ACKNOWLEDGMENT	iv
DEDICATION.....	v
LIST OF TABLES.....	x
LIST OF FIGURES	xi
ABSTRACT.....	xii
CHAPTER ONE	1
INTRODUCTION	1
1.0 Introduction.....	1
1.1 Background of the Study	1
1.2 Statement of the Problem.....	5
1.3 Purpose of the Study	6
1.4 Research Questions.....	6
1.5 Significance of the Study	7
1.6 Delimitation of the Study.....	7
1.7 Limitations	8
1.8 Definition of terms.....	8
1.9 Organisation of the Study	9
CHAPTER TWO	11
LITERATURE REVIEW	11
2.0 Introduction.....	11
2.1 Theoretical Review	11
2.1.1 Achievement Goal Theory.....	11
2.1.2 Social cognitive theory	15
2.1.3 The reciprocal-effects theory	19
2.2 Conceptual Review	21

2.3.1 Computer Programming Anxiety as a construct in Computer Science Education	21
2.2.3 Self-Efficacy in Computer Programming	22
2.2.4 Computer Programming Interest	23
2.2.4.1 Dimensions of Programming Interest	24
2.3 Empirical Review	25
2.3.1 Relationship between Computer Programming Anxiety and Achievement ..	25
2.3.2 Factors that Affect Computer Programming Anxiety.....	25
2.3.2.1 Academic Performance	26
2.3.2.2 Self-Efficacy	26
2.3.2.3 Correlates of Programming Anxiety	26
2.3.2.4 Debugging Challenges	26
2.3.2.5 Peer Interactions	26
2.3.2.6 Gender.....	27
2.3.2.7 Perceived Complexity.....	27
2.3.3 Students' Computer Programming Anxiety and Interest.....	27
2.3.3.1 Impact on Success.....	27
2.3.3.2 Development of Self-Beliefs	28
2.3.3.3 Lack of Confidence.....	28
2.3.3.4 Effect on Time and Practice.....	28
2.3.3.5 Impact on Debugging.....	28
2.3.4 Relationship between Self-Concept and Achievement.....	29
2.3.4.1 Self-Concept and Academic Achievement	29
2.3.4.2 Domain-Specific Self-Concept	29
2.3.4.3 Reciprocal Effects	29
2.3.4.4 Gender and Self-Concept.....	30
2.3.5 Students' Interest and Related Effect in Computer Programming Education	30
2.4 Conceptual Framework.....	31
CHAPTER THREE	33
METHODOLOGY	33
3.0 Introduction.....	33
3.1 Research Design	33

3.2 Population	33
3.3 Sample and Sampling Techniques	34
3.4 Data Collection Instrument	34
3.5 Data Collection Procedure	34
3.6 Reliability and Validity of the Instrument	34
3.7 Ethical Issues	35
3.8 Data Analysis Procedure	35
CHAPTER FOUR.....	36
RESULTS AND DISCUSSION	36
4.0 Introduction.....	36
4.1 Demographics	36
4.1.1 Gender.....	36
4.1.2 Age.....	36
4.2 Exploratory Factor Analysis (EFA)	36
4.3 Confirmatory Factor Analysis (CFA).....	38
4.4 Discriminant Validity	40
4.5 Path Analysis	41
4.5.1 Research Question 1: What is the effect of Anxiety on Computer Programming Achievement?	41
4.5.2 Research Question 2: What is the effect of Self-Concept (SC) on Computer Programming Achievement?	41
4.5.3 Research Question 1: What is the effect of Interest on Computer Programming Achievement?	42
4.6 Discussion of Results.....	44
4.6.1 The Effect of Anxiety on Computer Programming Achievement.....	44
4.6.2 The Effect of Self-Concept (SC) on Computer Programming Achievement	45
4.6.3 Relationship between Students’ Interest and Computer Programming Achievement	46

CHAPTER FIVE	49
SUMMARY, CONCLUSIONS AND RECOMMENDATIONS	49
5.0 Introduction.....	49
5.1 Summary	49
5.2 Conclusion	49
5.3 Recommendation	50
5.4 Suggestions for Further Studies	51
REFERENCES.....	52
APPENDIX.....	66

LIST OF TABLES

Table 1: Demographics	36
Table 2: Exploratory Factor Analysis (EFA).....	37
Table 3: Confirmatory Factor Analysis (CFA).....	39
Table 4: Discriminant Validity	41
Table 5: Path Analysis	41

LIST OF FIGURES

Figure 1: Bandura's triadic reciprocity	17
Figure 1: Conceptual Framework	32
Figure 2: Confirmatory Factor Analysis (CFA)	40
Figure 3: Path Summary	43

ABSTRACT

The poor performances of students in computer programming have been attributed to several factors over the years. Studies have attributed factors such as inadequate teaching personnel and resource, anxiety and fear of computer programming, poor teaching methods and interest of students to the poor performance of students in the subject. This study sought to investigate the effect of self-concept, student computer programming interest and computer programming anxiety on students' computer programming achievement. This study used a quantitative approach and adopted a descriptive survey design. A Likert scale questionnaire measuring all the constructs was given to 300 students who were selected using proportionate stratified and simple random sampling techniques from the Akyem Akenten Appiah-Menka University of Skills Training and Entrepreneurial Development in the Ashanti Region of Ghana. It was discovered that anxiety has no noticeable direct impact on computer programming achievement. There is no significant direct effect of Self-Concept on Computer Programming Achievement. It was found that there was a highly significant statistical relationship between Interest and Computer Programming Achievement. In other words, Interest has a strong direct impact on computer programming achievement. Therefore, it was suggested that computer programming instructors be encouraged to make computer programming classes engaging so that students will develop interest for the subject. Instructors should develop strategies for encouraging students to have positive self-concepts in computer programming.

CHAPTER ONE

INTRODUCTION

1.0 Introduction

The chapter one introduces the study and consists of the background, the problem statement, the purpose, the research questions, the research hypothesis, the significance, the delimitation, limitation, and organization of the study.

1.1 Background of the Study

Teaching programming courses is identified as one of the significant challenges in computer science education (Koulouri, Laurinen, & Repenning, 2015). Learning programming involves various aspects, including understanding language syntax and semantics, as well as developing problem-solving skills in general (Linn and Dalbey 1989). Therefore, most programming courses employ individual programming assignments to ensure that students learn and master programming skills (Hundhausen, Douglas, & Stasko 2015). Typically, in undergraduate programming courses, students are expected to complete content-specific assignments individually. This instructional approach serves several purposes. Firstly, it allows for a clear alignment between the learning objectives of the course and the programming assignments, ensuring that students are actively practicing and applying the concepts they have learned. Additionally, this approach helps to mitigate the risk of students falling behind in their understanding of the material if they happen to miss a particular assignment. By individually completing assignments, students can progress at their own pace and engage with the course content consistently, maximizing their learning outcomes in programming.

Self-concept refers to individuals' perception and evaluation of themselves, encompassing their beliefs, thoughts, and feelings about their own identity, abilities, and characteristics. It involves an individual's understanding of who they are, including their strengths, weaknesses, values, and beliefs about themselves (Harter, 2012; Shavelson, Hubner, & Stanton, 1976). The significance of the self-concept construct in education is widely recognized due to its association with academic success (Valentine, Dubois, & Cooper, 2004). According to Shavelson and Bolus (1982), self-concept is a multidimensional variable in education that encompasses both academic and non-academic aspects. Academic self-concept specifically pertains to how one perceives themselves in relation to their academic achievements (Reyes, 1984; Awan, Noureen, & Naz, 2011).

The role of self-concept is crucial in the educational process. When learners are acknowledged, affirmed, respected, and valued, they are more likely to develop a positive mindset of self-acceptance and self-esteem, which ultimately enhances their performance (Chowdhury & Pati, 1997). The relationship between self-concept and student achievement highlights the significance of self-concept as a construct in education (Byrne, 1984; Ayodele, 2011; Wang, 2007). Numerous studies have been conducted to explore the link between self-concept and academic performance, with a considerable number of them finding a strong positive correlation between self-concept and student achievement (Ayodele, 2011; Marsh & Hau, 2004; Wilkins, 2004; Wang, 2007; Yildiz & Fer, 2013).

Interest can be conceived as a construct which has effect on motivation and cognitive activity. Many studies have hypothesized that interest can lead to students' achievement in learning (Ainley, Hidi & Bendorf, 2002; Lee, Lee & Bing, 2014). These studies suggest

so on the grounds that interest leads to a positive mood during instruction, sustaining of attention, processing for directed goals, self-regulated process and deep levels of learning (Ainley, Hidi, & Berndorf, 2002; Lee, Lee, & Bing, 2014; Richardson & Suinn, 1972). Interest is an internal factor that has an impact on students' learning. When students are interested in a subject, they are motivated to spend time learning and engaging with tasks related to that subject. Interest can be understood as both a dispositional state, where individual interest persists, and an actualized state, where interest is demonstrated through behaviors such as focused attention, pleasure, and persistence in tasks (Krapp, Hidi, & Renninger, 1992). Educators aspire to see students with such dispositions towards learning.

Numerous studies have established a strong connection between students' academic interest and their academic achievement. Schiefele (1992) found a positive relationship between interest and knowledge across multiple studies. Higher levels of interest have been shown to predict increased attention, engagement, perseverance, perceived competence, and the use of effective learning strategies, all of which are crucial factors for successful learning (Ainley, Hillman, & Hidi, 2002). Interest can also serve as a predictor of students' achievement and act as a mediator between achievement and anxiety.

Anxiety, a pervasive emotional state, has garnered increasing attention in educational research due to its potential impact on academic performance. As students navigate the challenges of academia, the weight of expectations, social pressures, and the fear of failure can give rise to various forms of anxiety. Anxiety disrupts attention and concentration, leading to difficulties in maintaining focus during academic tasks. According to Flett, Hewitt, Boucher, Davidson & Munro, 2016 and Pariante, 2006,

anxiety can hinder creative thinking and problem-solving skills, negatively impacting academic achievement. Anxiety can impede interpersonal interactions, affecting classroom participation and collaborative learning (Inderbitzen-Nolan & Walters, 2000). Hembree (1988) found that heightened anxiety correlates with decreased performance on cognitive tasks and can hinder information recall.

The realm of computer programming is characterized by a multifaceted interplay of student interests, self-concept, and anxiety. Students' fascination with coding, software development, and problem-solving using programming languages is marked by natural attraction, curiosity, and passion. This intrinsic motivation leads them to explore coding tasks, gain knowledge, and persevere through challenges. This dynamic enthusiasm significantly influences their engagement, educational achievements, and career aspirations (Dargahi & Adams, 2016). Intrinsically motivated students find programming inherently enjoyable and satisfying, engaging in it for personal fulfillment. Within the domain of programming, self-concept plays a pivotal role, combining the ability to independently tackle challenges with the necessary confidence and motivation. Various personal factors, such as desired outcomes, motivation, and objectives, profoundly impact student engagement (Elteгани & Butgereit, 2015; Yilmaz & Koc, 2021). Students often encounter difficulties with complex programming and algorithmic issues, and those with low self-concept struggle due to a lack of effort and determination to overcome obstacles. On the flip side, computer programming anxiety is a psychological construct marked by fear and uneasiness when students engage in programming tasks (Jenkins, 2019). This anxiety can hinder comprehension of coding concepts and completion of assignments, affecting overall learning outcomes (Fortune & Good, 2017). Students grappling with programming anxiety may struggle to maintain motivation in their

computer science courses, potentially diminishing their intrinsic motivation to explore programming and persist through coding challenges (Newby, 2019).

While the effect of self-concept, interest and programming anxiety have been studied globally in computer programming, there is a dearth of research specifically on these constructs in Ghana. This study examines the effect of students' computer programming self-concept, interest and computer programming anxiety on achievement.

1.2 Statement of the Problem

Despite the increasing importance of computer programming skills in various fields, there is a growing concern regarding students' achievement in computer programming courses. Research has confirmed that students enrolled in computer programming courses often face difficulties with one or more of the essential concepts required for success. According to a study conducted by Beise, Myers, VanBrackle, and Chevli-Saroq in 2003, the likelihood of passing a first introductory programming course on the first attempt was 40% for students in all majors. The study revealed an initial failure rate of 19.5% and a withdrawal rate of 40.5%. In a comprehensive review by Watson and Li in 2014, it was reported that the average pass rate for introductory computer programming courses worldwide is approximately 67.7%. This study encompassed 161 outcomes from introductory programming courses across 15 countries and 51 different academic institutions. These statistics continue to be a significant concern for computing faculty around the globe.

In the field of computer programming education, several factors have been identified as having an impact on how well students perform. These factors include students' anxiety towards computer programming, their level of interest in the subject, and their perception of themselves in relation to computer programming, which can hinder their learning and performance. While many studies have explored the connection between teacher

motivation, teaching strategies, and students' performance in computer programming, there have been few studies conducted in Ghana that specifically examine how students' interest in computer programming, their self-concept and their computer programming anxiety affect their computer programming achievement. To address this gap in knowledge, this study focused on investigating the roles of students' computer programming interest, self-concept and computer programming anxiety in their overall computer achievement among the Information Technology students in the Akenten Appiah-Menka University of Skills Training and Entrepreneurial Development (AAMUSTED).

1.3 Purpose of the Study

The aim of this study was to investigate how students' self-concept, interest in computer programming and computer programming anxiety affect academic achievement among Information Technology students at AAMUSTED. In order to achieve the purpose, the following specific objectives guided the study:

1. To determine the effect of Information Technology students' computer programming anxiety on their computer programming performance.
2. To determine the effect of Information Technology students' computer programming self-concept on their computer programming achievement.
3. To determine the effect of Information Technology students' computer interest on their computer programming achievement.

1.4 Research Questions

The following questions were posed to guide the study:

1. What is the effect of Information Technology students' computer programming anxiety on their performance in computer programming?

2. What is the effect of Information Technology students' computer programming self-concept on their achievement in computer programming?
3. What is the effect of Information Technology students' computer programming interest on their achievement in computer programming?

1.5 Significance of the Study

The outcomes of this study will contribute to educators' understanding of the multifaceted impact of students' computer programming anxiety, self-concept, and computer programming interest on their performance. It will increase teachers' awareness of how these behavioral factors influence students' computer programming achievement. The study's findings regarding the effect of self-concept, students' computer programming interest and computer programming anxiety on student computer programming achievement will enhance teachers' comprehension of how these factors interact to influence students' performance.

This research will offer a fresh perspective on the existing body of knowledge regarding computer programming anxiety. It will illuminate the significance of self-concept and computer programming interest in the context of computer programming anxiety and its effects on achievement.

1.6 Delimitation of the Study

This study focused on examining how self-concept, computer programming interest and computer programming anxiety affect students' computer programming achievement. The study investigated the variables of computer programming anxiety, computer programming self-concept, computer programming interest as independent variables, and computer programming achievement as the dependent variable. The research was conducted exclusively among Information Technology students at Akenten Appiah-

Menka University of Skills Training and Entrepreneurial Development (AAMUSTED) in the Ashanti region of Ghana.

Methodologically, the study employed a quantitative approach and utilized questionnaires as the primary data collection instrument to gather information from the students regarding the different variables. The relationships among these constructs were analyzed using a Structural Equation Model.

1.7 Limitations

As this study was specifically conducted at AAMUSTED, it is important to note that the findings cannot be applied universally to represent students' success in other regions of the country. Therefore, the study does not make any claims or attempts to generalize its findings beyond the scope of the research conducted at AAMUSTED.

1.8 Definition of terms

The following are the definitions of some constructs in the study:

Computer programming anxiety

Computer programming anxiety refers to the psychological and emotional state of unease, apprehension, or fear experienced by individuals when engaging in computer programming tasks or activities. It is characterized by feelings of stress, self-doubt, or worry related to programming concepts, problem-solving, coding, debugging, or the overall process of writing and executing code. Computer programming anxiety can manifest as performance anxiety, fear of failure, or a lack of confidence in one's programming abilities. It may hinder an individual's productivity, learning, and overall enjoyment of computer programming tasks.

Computer programming interest

Computer programming interest refers to an individual's inclination, curiosity, and enthusiasm towards the field of computer programming. It involves a genuine attraction and fascination with the concepts, principles, and practices of programming. A person with computer programming interest may exhibit a strong desire to explore, learn, and engage in programming activities. They may find enjoyment and satisfaction in problem-solving, coding, designing algorithms, and creating software applications. Computer programming interest often drives individuals to actively seek opportunities to expand their knowledge and skills in programming, and it can contribute to their motivation, perseverance, and success in the field.

Self-Concept

Self-concept refers to an individual's perception, beliefs, and evaluation of themselves. It encompasses their thoughts, feelings, and evaluations of their own characteristics, abilities, values, and identity. It includes an understanding of one's strengths, weaknesses, personality traits, and overall sense of self. The self-concept plays a crucial role in shaping one's self-esteem, self-confidence, and behavior in various domains of life.

1.9 Organisation of the Study

The study consists of five main chapters. The introductory chapter provides an overview of the study, including the background, problem statement, purpose, research objectives, and hypothesis. It also covers the delimitations, significance of the study, and the organization of the research.

The second chapter, known as the literature review, encompasses the theoretical review, empirical review, and conceptual review. It explores relevant literature, theories, and studies that are essential to understanding the subject matter. Additionally, the conceptual

framework of the study is presented in this chapter. Chapter three focuses on the methodology employed to gather the necessary data. It details the research design, population, sample, and sampling techniques. The chapter provides in-depth information on the data collection instrument, including its procedures, validity, reliability, and ethical considerations. Furthermore, it discusses the chosen data analysis approach. In chapter four, the results of the study are presented using tables and figures. The findings are then thoroughly discussed and analysed. Chapter five concludes the study and includes a summary of the research, key conclusions drawn from the findings, and recommendations based on the results. It also suggests areas for further research or study.

CHAPTER TWO

LITERATURE REVIEW

2.0 Introduction

The study sought to investigate the effect of Self-Concept, Students' Computer Programming Interest and Computer Programming Anxiety on Students' Achievement. This chapter contains a review of the relevant literature in this study. The literature review is divided into three sections: theoretical review, empirical review, and conceptual review. This chapter also includes the study's conceptual framework.

2.1 Theoretical Review

2.1.1 Achievement Goal Theory

The concept of achievement theory is prevalent not only in the sports arena but also in academic achievement within schools, as highlighted by Üztemur (2020). Present-day society places great emphasis on the performance of students, leading to various rankings and competition among schools. Parents and guardians now have high expectations for their children's academic performance, which influences their choice of schools. In the past, people engaged in sports for the sheer enjoyment of it, and students pursued courses out of curiosity about specific subjects. However, this narrative has shifted over time (Üztemur, 2020).

A sociocognitive theory known as the achievement goal theory seeks to elucidate how students approach academic tasks, their motivation to complete them, and the underlying motivations behind their behavioral patterns (Elliott & McGregor, 2001; Zhou et al., 2019). Initially, goal theorists distinguished between two types of goal orientations: mastery-coping and performance goal orientations (Dweck & Leggett, 1988). Subsequently, research has further categorized performance goal orientations into two

sub-dimensions: performance approach and performance avoidance (Ames, 1992; Elliot & Harackiewicz, 1996). Mastery goal orientation involves a student's efforts to develop competencies, gain a comprehensive understanding, and advance without the need for external rewards (Duman & Eren, 2014). Striving to acquire new skills and self-improvement driven by intrinsic motivation characterizes a mastery-oriented goal for students (Üztemur, 2020). Such students rely on their internalized norms rather than external authorities, like teachers, parents, or peers, to gauge their learning progress (Jagacinski & Strickland, 2000).

However, students with a performance-approach goal aim to outperform their peers, seeking to prove themselves and engage in friendly competition (Elliot & Harackiewicz, 1996). This group of students desires to gain the approval of others and therefore tends to avoid situations where they are uncertain about how they may be perceived by their peers. Consequently, they strive to excel and make a positive impression on those around them. Consequently, they become disheartened in situations where they are judged negatively due to their failures, leading them to avoid such circumstances and leave their work incomplete or undone (Ozgunor, 2014; Cetin et al., 2014; Vandewalle, 1997).

The socio-cognitive theory known as achievement goal theory seeks to elucidate how students approach academic tasks, their level of motivation to complete them, and the underlying motivations driving their behavior (Elliott & McGregor, 2001; Zhou et al., 2019). Initially, goal theorists distinguished between two primary goal orientations: mastery-coping and performance goal orientations (Dweck & Leggett, 1988). Subsequently, performance goal orientations were further classified into two sub-dimensions: performance approach and performance avoidance (Ames, 1992; Elliot & Harackiewicz, 1996). Mastery goal orientation involves a student's efforts to develop

competencies and gain a broad understanding without relying on external rewards (Dweck & Leggett, 1988).

The development of goal theory was influenced by earlier work on needs and motives by McClelland and Atkinson (Maehr & Zusho, 2009). Many of the fundamental concepts in contemporary goal frameworks have their roots in the study of motives (Elliot, 1999). Some theorists have suggested that achievement goals represent two facets of achievement motives, characterized as a motivation to succeed and a motivation to avoid failure (Elliot and Church, 1997). In the history of goal theory, attribution theory has had a significant impact. Maehr and Nicholls (1980) explain the crucial role of future expectations in goal theories. Goal theory provides insights into why students with similar academic abilities may react differently to the same academic tasks and why some display signs of helplessness while others exhibit signs of mastery (Dweck & Leggett, 1988; Elliott & Dweck, 1988).

An exploration of the literature on goal theory reveals the existence of two main perspectives: the traditional viewpoint and the normative viewpoint (Maehr & Zusho, 2009). Presently, this is often referred to as the multiple goals viewpoint, which has been popularized by researchers such as Harackiewicz, Elliot, and their colleagues (Harackiewicz et al., 2002; Midgley, Kaplan, & Middleton, 2001; Roeser, 2004). Maehr and Zusho (2009), however, offer an alternative view of goal theory that encompasses various aspects, including the origin of goals, the issue of motivational value, the role of performance goals, and the attractiveness of goals, even in their aversive nature, as well as the potential for multiple goals coexisting.

In most cases, these various viewpoints recognize the bidirectional impact of goal theory as it represents a manifestation of the social-cognitive approach to motivation.

In the context of computer programming achievement, the Achievement Goal Theory can be applied to understand how individuals approach and perceive their goals in learning and mastering programming skills. Individuals with a mastery orientation in computer programming will seek to develop and master coding skills. They will be motivated by the desire to understand programming concepts thoroughly. Also, some individuals in computer programming may be driven by the desire to outperform their peers, seeking positive recognition for their coding abilities.

The application of the Achievement Goal Theory in computer programming can have several implications:

Learning Environment: Educators can create a positive learning environment that encourages a mastery-oriented approach, emphasizing the development of programming skills and understanding rather than merely competition.

Feedback and Assessment: Feedback mechanisms can be designed to focus on growth and improvement, supporting individuals with mastery goals. Encouraging constructive feedback helps learners see mistakes as opportunities for learning.

Collaboration: Emphasizing collaborative learning can foster a positive atmosphere, where individuals share knowledge and work together to achieve mastery rather than viewing peers as competitors.

Understanding the achievement goals of individuals in computer programming can inform instructional strategies, promote a growth mindset, and contribute to the creation of an environment conducive to sustained learning and achievement in this technical field.

2.1.2 Social cognitive theory

The social cognitive learning theory is based on the idea that individuals learn by observing and imitating the actions of those around them, and it emphasizes the critical role of human interaction in shaping one's identity. Bandura, in the mid-1980s, adopted a more comprehensive approach to this theory, focusing on providing a deeper understanding of human cognition within the context of social learning (Nabavi, 2012). Bandura's work expanded the initial social learning theory and subsequently gave rise to the social cognitive theory (Bandura, 1999), which provides a framework for comprehending, predicting, and modifying human behavior (Green & Piel, 2009).

Social cognitive learning places a greater emphasis on cognitive aspects, particularly on how both adults and children think during social interactions and how this thinking influences their behavior and development. Nabavi (2012) suggests that Bandura's rejection of behaviorism and his emphasis on the social antecedents of social cognitive processes contributed to the theory's integration into the broader paradigm of developmental theories. In essence, Bandura argued that people acquire both behavioral and cognitive skills by observing others' behavior, and learning can occur without the need for active reinforcement. The central idea of the social cognitive theory is that individuals are capable of self-regulating their thoughts, feelings, motivations, and actions (Nabavi, 2012). Self-regulation, in this context, is the process by which an individual control and coordinate his actions. It involves individuals actively creating adaptive patterns of thinking and behaving in response to the environmental conditions required to achieve specific goals. Effective self-regulation may involve a cyclical process in which individuals plan, develop workable strategies, implement those strategies, and then assess their performance outcomes (Locke & Latham, 1990). The social learning theory posits that self-regulation occurs when individuals are aware of

what behaviors are acceptable or unacceptable and choose their behavior accordingly (Williams, 2010). Bandura (1978) emphasized the importance of promoting self-regulation and believed that individuals should be encouraged to reward themselves after performing necessary behaviors.

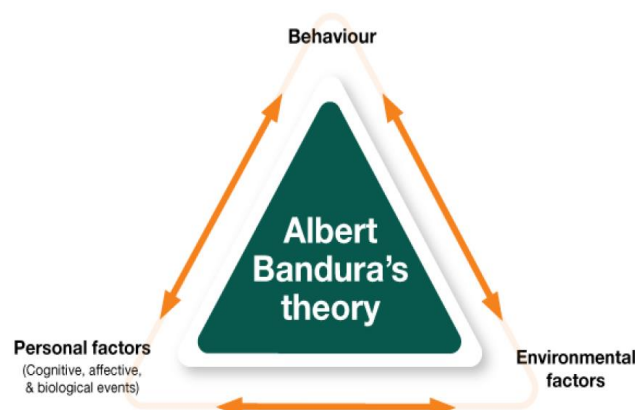
The bidirectional relationship between behavior and environmental conditions makes individuals both products and producers of their environment (Nabavi, 2012). They are influenced by the factors that shape and create the environment they experience, and, based on their acquired tendencies and competencies, individuals tend to select partners and activities from the numerous possible options available (Bandura & Walters, 1959; Bullock & Merrill, 1980; Emmons & Diener, 1986). An individual's behavior determines which potential environmental consequences are actualized and the forms they take, while the environmental consequences, in turn, influence which behavioral patterns are generated and enacted. The social learning theory suggests that this process remains consistent throughout life, as individuals of any age can engage in observational learning. According to this theory, people can learn from each other by observing, imitating, and modeling (Nabavi, 2012). It also indicates that learning can occur without an immediate change in behavior (Bandura, 1965), and a change in behavior may or may not result from learning (Bandura, 2006a). Over time, Bandura has demonstrated that cognition plays a significant role in learning, leading social learning theory to take on a more cognitive interpretation of human learning (Newman & Newman, 2015).

Research supports the notion that a substantial portion of behaviors can be learned through modeling, as individuals observe and replicate behaviors exhibited by others. Examples include children watching their parents' study, demonstrations of solving arithmetic problems, or individuals displaying courageous or fearful behavior in response

to certain situations (Bandura, 2006b; Nabavi, 2012). Another form of learning is associated with intrinsic rewards, such as experiencing satisfaction, fulfillment, and a sense of achievement (Nabavi, 2012). Some studies, like those by Muro and Jeffrey (2008), corroborate the principles of the social cognitive theory, which places a greater emphasis on internal thoughts and cognitions, bridging learning theories with cognitive developmental theories. Bandura (1986) later acknowledged that factors beyond external, environmental reinforcement can influence learning and behavior. People are not solely products of their circumstances; they actively create favorable conditions and take control of them. Their deliberate choices regarding the circumstances may have a substantial impact on their outcomes (Bandura, 1997). Bandura described triadic reciprocity as a central principle with three interdependent components (Nabavi, 2012). Many studies in this field support Bandura's theory of triadic reciprocity, which explains human behavior as the interaction of the environment, behavior, and individual factors in a dynamic and complementary manner.

According to Nabavi (2012), these are reciprocal in nature, as the figure 1 below illustrates.

Figure 1: *Bandura's triadic reciprocity*



Note. This theory was produced by Albert Bandura in 1977. From “Bandura’s Social Learning Theory & Social Cognitive Learning Theory” By Nabavi, 2012.

Within the informative model of human behavior in social cognitive theory, self-efficacy is a significant concept, and it causally influences the desired outcome of behavior but not the other way around (Bandura, 1986c, 1995, 1998, 2006b). Beliefs about self-efficacy have various effects on cognitive, motivational, emotional, and decision-making processes (Nabavi, 2012). Whether an individual tends to think in a pessimistic or optimistic manner, or in a self-enhancing or self-debilitating way, depends on their self-efficacy beliefs. Self-efficacy plays a crucial role in self-regulating motivation when facing challenges and striving for desired results (Mark & Campbell, 2011). It is a fundamental component of the social cognitive learning theory and represents an individual's perceptions of their ability to successfully perform a specific behavior.

One of the key aspects of self-efficacy theory involves identifying four sources of efficacy information and learning experiences that contribute to self-efficacy expectations (Nabavi, 2012). According to Bandura's theory, researchers like Betz (2007) and McCormick & Martinko (2004) have supported the idea that self-efficacy can influence behavior and cognition in several ways, including choice of actions, goal setting, effort, learning, and achievement determination. Individuals with low self-efficacy tend to avoid challenging tasks and often focus on their own limitations and negative outcomes, whereas those with high self-efficacy are more inclined to view challenging tasks as opportunities to excel rather than avoid (Mark & Campbell, 2011).

In the context computer programming achievement, this theory suggests that individuals learn and acquire programming skills not only through personal experience but also by observing the behavior and outcomes of others. For instance, individuals can learn programming skills by observing and emulating successful programmers. This might include watching coding tutorials, studying well-crafted code, or participating in

collaborative coding projects where they can observe effective problem-solving strategies. Social Cognitive Theory supports the idea that learning is a social process. Collaborative learning in computer programming, such as pair programming or group projects, allows individuals to observe, learn from, and collaborate with peers, fostering a sense of community and shared knowledge. In summary, Social Cognitive Theory suggests that individuals in computer programming can benefit from observing, modeling, and collaborating with others. Creating a supportive learning environment that includes positive role models, collaborative opportunities, and effective feedback mechanisms can enhance the achievement and learning experiences in computer programming.

2.1.3 The reciprocal-effects theory

The relationship between academic success and self-concept has been a topic of discussion, with questions arising about whether one predicts the other, or if it works both ways. In the past, self-enhancement theorists argued that, in an educational context, enhancing one's self-concept is crucial for improving academic achievement (Calsyn & Kenny, 1977; Shavelson & Bolus, 1982). However, theories of skill development suggested that enhancing academic achievement actually boosts self-concept (Calsyn & Kenny, 1977; Skaalvik & Valas, 1999). In light of these contrasting views, if self-concept leads to improved achievement, then teachers should focus on strategies that enhance self-concepts. Conversely, if achievement leads to an improved self-concept, then instructors should concentrate on improving academic performance. This approach has been described as a "classical causal ordering study" by Marsh and Craven (2005, p. 25). The complementary impact model posits that prior academic achievement is linked to the development of academic self-concept in the future, and earlier academic self-concept is related to the enhancement of academic achievement.

Researchers have not reached a consensus on how these components interact in the context of the relationship between self-concept and academic success. There are three distinct models used to compare self-concept and academic achievement: the self-enhancement model, the achievement development model, and the reciprocal-effects model.

1. The self-enhancement model suggests that a student's success in school is dependent on positive self-concepts, indicating that it is more effective to boost self-concept than to focus on improving achievement (Green et al., 2006).

2. The achievement development model, on the other hand, argues that the best way to enhance self-concept is by building academic success and confidence.

3. The reciprocal-effects model can be seen as a compromise between the two models. In this model, academic skills and success contribute to the development of self-concept, and self-concept, in turn, has a positive impact on achievement. These two factors mutually influence each other (Green et al., 2006). The reciprocal-effects paradigm posits that prior academic success or failure influences one's self-concept, which then influences future academic performance either positively or adversely.

In the context of computer programming achievement, the theory could mean that a learner's skills and behavior are influenced by their experiences and interactions within the programming environment. A learner's existing programming skills, intrinsic motivation, and self-efficacy beliefs can influence their approach to coding tasks and their willingness to persist in the face of challenges. The actions and behaviors of individuals in programming, such as their coding practices, problem-solving strategies, and effort invested in learning, can impact their overall achievement. Since the theory puts premium on the environment, it can be inferred that, the programming environment, including the nature of coding assignments, the availability of resources, and the support from peers

and instructors, can shape the learner's experiences and achievements. The theory suggests that personal factors, behavior, and the environment are in constant interaction. For example, a positive learning environment with supportive peers and instructors can enhance a learner's motivation, leading to improved coding practices and better outcomes. Conversely, positive outcomes and successful coding experiences can further boost motivation and self-efficacy.

2.2 Conceptual Review

2.3.1 Computer Programming Anxiety as a construct in Computer Science Education

Computer Programming Anxiety is an emerging construct in the field of Computer Science Education. This conceptual review aims to provide an overview of Computer Programming Anxiety, its definitions, and the implications it holds for students and educators in the context of computer science education.

Computer Programming Anxiety can be defined as a psychological construct characterized by the fear, apprehension, and uneasiness that students experience when engaging in computer programming tasks, assignments, or projects (Jenkins, 2019). It is often linked to the perceived complexity of coding, the fear of making errors, and the pressure to meet academic expectations. This construct is akin to other forms of performance anxiety but is specific to the context of programming.

Programming anxiety can hinder students' ability to grasp coding concepts and complete assignments, which can affect their learning outcomes (Fortune & Good, 2017). Students experiencing programming anxiety may struggle to stay motivated in their computer science courses. This construct can diminish students' intrinsic motivation to explore programming and persevere in the face of coding challenges (Newby, 2019). Research indicates that programming anxiety may contribute to gender disparities in computer

science education. Female students are more likely to experience anxiety related to programming, which can discourage them from pursuing computer science careers (Beyer, 2020). Prolonged programming anxiety can have negative effects on students' mental well-being. It can lead to stress, frustration, and a lack of self-confidence, impacting not only their academic performance but also their overall quality of life (Smith, 2021).

Several factors contribute to the development of programming anxiety. The intricate nature of coding languages and the fear of not being able to comprehend them can be intimidating for novice programmers (Jenkins, 2019). Students often fear making mistakes in their code, which can lead to programming anxiety (Fortune & Good, 2017). Students may feel the pressure to perform well in a field where peers or society expect high achievement, which can exacerbate anxiety (Newby, 2019). Inadequate support and resources, including teacher guidance and relevant programming tools, can contribute to programming anxiety (Beyer, 2020).

2.2.3 Self-Efficacy in Computer Programming

Self-efficacy, an active precursor to the development of one's self-concept (Bong & Skaalvik, 2003), pertains to the capacity to tackle and resolve problems or tasks, involving the ability to effectively plan and execute the required activities to attain desired outcomes (Zwart et al, 2020). It signifies a person's confidence in their own abilities, particularly their achievement to perform unfamiliar or challenging tasks, and is intricately intertwined with motivation. In essence, self-efficacy reflects an individual's belief in their capability to achieve success. Research indicates that education structured around tasks can have a notably positive impact on self-efficacy (Garaika et al, 2019). Self-efficacy plays a pivotal role in the realms of learning, accomplishment, and

adaptability to emerging technologies. It is widely acknowledged as the linchpin to triumph in various endeavors, including the process of learning how to program.

Individuals with elevated self-efficacy hold greater expectations for themselves, invest more effort in surmounting obstacles, and, as a result, exhibit enhanced competence in addressing intricate challenges (Bandura, 2012). Self-efficacy beliefs significantly influence whether individuals adopt self-affirming or self-undermining thought patterns, the degree of motivation they exhibit, and their capacity to persevere effectively when confronted with adversity (Schwarzer & Warner, 2013).

In the realm of programming, self-efficacy involves combining the ability to independently solve programming challenges with the self-assurance and motivation needed for such endeavors. Several individual factors are instrumental in driving student engagement, encompassing desired outcomes, motivation, and objectives (Elteгани & Butgereit, 2015) (Yilmaz & Koc, 2021). It is not uncommon for students to encounter challenges when grappling with intricate programming and algorithmic issues, and those with low self-efficacy often falter due to a lack of effort and determination to surmount these hurdles.

2.2.4 Computer Programming Interest

Computer programming interest can be described as the inclination, fascination, and enthusiasm that students exhibit toward the process of coding, software development, and problem-solving using programming languages. It encompasses their motivation to explore coding tasks, their eagerness to acquire more knowledge about programming, and their determination to persevere when confronted with challenges. Interest in computer programming is a dynamic and multifaceted construct that significantly

influences students' engagement, learning outcomes, and career aspirations (Dargahi & Adams, 2016).

Deci & Ryan 2000, suggests that individuals have innate psychological needs for autonomy, competence, and relatedness. When students find programming inherently enjoyable and fulfilling, their interest is considered intrinsically motivated. In this context, students engage in programming because they derive personal satisfaction from it. The Four-Phase Model of Interest Development (Hidi & Renninger, 2006) outlines how interest evolves over time. It commences with situational interest, where external factors spark curiosity. If this initial interest is nurtured, it can develop into individual interest, becoming a more stable and enduring personal interest. For programming, situational interest might be triggered by an engaging coding project, which, if sustained, can lead to a long-term passion for coding.

2.2.4.1 Dimensions of Programming Interest

Affective Interest: This dimension relates to the emotional component of interest. Students with affective interest in programming experience positive emotions, such as excitement and enjoyment, when engaging in coding activities (Eijkelhof, 2014).

Cognitive Interest: Cognitive interest pertains to the intellectual curiosity and fascination that students have for programming concepts and problem-solving. It drives them to explore and comprehend coding principles (Bergin, 1999).

Behavioral Interest: Behavioral interest encompasses students' actions and engagement with programming. It includes their commitment to completing coding assignments, participation in coding-related extracurricular activities, and the time they invest in programming (Margulieux, Guzdial & Catrambone, 2012).

Students' interest in programming has significant implications for their learning outcomes, career choices, and problem-solving skills. It positively influences their academic performance and motivates them to pursue computer science-related careers, helping to address the skills gap in the technology industry.

2.3 Empirical Review

2.3.1 Relationship between Computer Programming Anxiety and Achievement

In the past decade, there has been a significant increase in the number of studies focusing on programming anxiety. Some of these studies have delved into the factors associated with programming anxiety, while others have explored its impact on student performance and motivation. According to research by Sinožić and Orehovaki (2018), several key factors contribute to programming anxiety among novices, including the lack of programming experience, fear of programming, and the misconception that programming languages are highly complex. Similarly, unfamiliar topics within programming courses tend to deter students from engaging with programming and make them feel uneasy, as reported by Olipas et al. (2021). These studies indicate that students' levels of programming anxiety tend to increase as they are introduced to new programming concepts and principles (Campbell, 2018; Dasuki & Quaye, 2016). This body of research sheds light on the challenges faced by students and the psychological factors that can impact their programming experiences.

2.3.2 Factors that Affect Computer Programming Anxiety

Programming anxiety has been the subject of numerous studies, and researchers have identified several factors and correlations that are associated with it. These factors include academic performance, self-efficacy, encountering errors during program development,

gender, peer interactions, test anxiety, computer anxiety, mathematics, and debugging challenges.

2.3.2.1 Academic Performance

Research, such as the study by Olipas et al. (2021), has revealed a negative association between programming anxiety and academic performance. High levels of programming anxiety tend to be linked to lower academic achievement.

2.3.2.2 Self-Efficacy

Self-efficacy plays a crucial role in performance in programming courses, as explained by Hsu and Gainsburg (2021) and Wilfong (2006). It acts as a mediating factor in the relationship between anxiety and performance.

2.3.2.3 Correlates of Programming Anxiety

A systematic review of the literature conducted by Nolan and Bergin (2016) found that several factors correlate with programming anxiety. These include programming as a subject, test anxiety, computer anxiety (related to the volume of computer usage), and the frequent use of mathematics in coding.

2.3.2.4 Debugging Challenges

Students' inability to debug their programs is another factor that can increase programming anxiety, as indicated by studies like Dasuki & Quaye (2016) and Nolan and Bergin (2016).

2.3.2.5 Peer Interactions

Some researchers, including Nolan and Bergin (2016) and Falkner, Falkner, and Vivian (2013), have highlighted the effects of peer interactions on programming anxiety.

Learning to program in a laboratory with many peers can be stressful, and collaborative practices can sometimes create fear and tension in learners.

2.3.2.6 Gender

Gender also plays a role in programming anxiety, with female students often reporting higher levels of anxiety than male students, as shown in the study by Olipas and Luciano (2020).

2.3.2.7 Perceived Complexity

There is a strong association between the perceived complexity of programming tasks and programming anxiety. As programming assignments are perceived as more complex, students tend to report higher levels of anxiety, according to research by Chang (2005). These studies provide valuable insights into the various factors and relationships that impact programming anxiety, helping educators and researchers better understand how to support students in computer science and programming courses.

2.3.3 Students' Computer Programming Anxiety and Interest

Programming anxiety is recognized as a significant factor that can lead to students struggling and losing interest in programming courses. Several studies in the literature emphasize the critical role of programming anxiety in determining students' success and their engagement in programming courses.

2.3.3.1 Impact on Success

Researchers such as Connolly et al. (2007), Figueroa & Amoloza (2015), Kinnunen & Malmi (2006), Nolan, Bergin & Mooney (2019), Owolabi et al. (2014), and Scott (2015) have all highlighted the impact of programming anxiety on students' success in programming courses.

2.3.3.2 Development of Self-Beliefs

Kinnunen and Simon (2012) argue that students' self-beliefs are shaped by their experiences in programming activities rather than the quality of the programs they produce. Negative experiences and self-assessments can deter students from dedicating time and motivation to programming (Kinnunen & Malmi, 2006; Scott, 2015).

2.3.3.3 Lack of Confidence

Maguire et al. (2017) suggest that programming anxiety leads to a lack of confidence and discourages students from independently engaging in programming.

2.3.3.4 Effect on Time and Practice

Özmen and Altun's study (2014) found that students with lower levels of programming anxiety tend to spend more time on programming and produce higher-quality programs. Conversely, those with high anxiety levels dedicate less time to programming and avoid learning programming. Scott (2015) also supports this, concluding that programming anxiety hinders the time spent on programming practice and reduces course participation (Bergin & Reilly, 2005).

2.3.3.5 Impact on Debugging

Scott and Ghinea (2014) conducted a study on the adverse effects of programming anxiety on students' programming practice. The findings revealed that students frequently experience concerns when undertaking debugging activities.

These findings collectively underscore the detrimental impact of programming anxiety on students' performance, engagement, and overall experiences in programming courses. It emphasizes the importance of addressing programming anxiety to support students in their programming endeavours.

2.3.4 Relationship between Self-Concept and Achievement

2.3.4.1 Self-Concept and Academic Achievement

Numerous studies have explored the connection between self-concept and academic achievement. A study by Marsh and Craven (2006) found that students with high academic self-concept tend to perform better academically. This positive relationship between self-concept and achievement has been confirmed in various educational settings and across different age groups (Hattie, Marsh, Neill, & Richards, 1997; Valentine, DuBois, & Cooper, 2004).

2.3.4.2 Domain-Specific Self-Concept

Research suggests that the relationship between self-concept and achievement can be domain-specific. Shavelson, Hubner, and Stanton (1976) introduced the concept of domain-specific self-concept, highlighting that students may have different self-concepts in various academic subjects (e.g., math, science, language arts). Findings by Valentino and Stigler (2018) support this, showing that domain-specific self-concept is a better predictor of achievement in the respective subject areas.

2.3.4.3 Reciprocal Effects

The relationship between self-concept and achievement is not unidirectional. Some studies propose a reciprocal relationship, suggesting that high academic achievement can enhance self-concept (Möller & Marsh, 2013). This reciprocal relationship underscores the importance of nurturing both positive self-concept and academic success simultaneously.

2.3.4.4 Gender and Self-Concept

Gender differences in self-concept and its impact on achievement have also been examined. A study by Hyde, Fennema, and Lamon (1990) found that gender stereotypes and societal expectations can influence the self-concept of girls in math-related fields, affecting their math achievement. However, research by Stoet and Geary (2018) indicates that gender differences in self-concept have diminished over time, with girls achieving more parity in math self-concept and performance.

2.3.5 Students' Interest and Related Effect on Computer Programming Education

Several studies have investigated the relationship between students' interest in computer programming and their learning outcomes. For example, Baldi and Goodman (2020) conducted a study that found a positive correlation between students' interest in programming and their ability to grasp complex coding concepts. Likewise, Eren and Eryilmaz (2019) discovered that students who were more interested in programming tended to achieve higher grades in computer science courses. These findings suggest that a strong interest in programming can contribute to improved learning outcomes and academic success.

Students' interest in programming is closely linked to their motivation to engage with the subject. A study by Simpson and Huggins (2018) revealed that students who found programming interesting were more likely to persevere when faced with challenging coding tasks. This intrinsic motivation, driven by interest, can lead to higher levels of engagement, increased effort, and a deeper understanding of programming concepts (Deci & Ryan, 2000).

Interest in computer programming can also have a lasting impact on students' career choices. Research by Lin and Ren (2021) indicated that students who developed a strong

interest in programming during their educational journey were more likely to pursue careers in computer science, software development, or related fields. In contrast, students with low interest may opt for different career paths. This suggests that fostering interest in programming education can positively influence future career decisions.

Several studies have explored the factors that influence students' interest in computer programming. One key factor is the quality of teaching and instructional methods. Research by Brophy (2005) demonstrated that teachers who use engaging and interactive teaching techniques can stimulate students' interest in programming. Additionally, the availability of interesting and relatable coding projects, such as app development or game design, can pique students' interest (Papastergiou, 2009).

Gender differences in programming interest have also been a subject of investigation. Studies by Vekiri (2010) and Margolis et al. (2008) revealed that there is often a gender gap in programming interest, with boys showing more interest than girls. However, interventions that create an inclusive and welcoming learning environment can mitigate these differences (Tissenbaum, Slotta, & Lui, 2019).

2.4 Conceptual Framework

In this study, the researcher hypothesized that, the effect that computer programming anxiety (AN) has on students' computer programming achievement (CPA) can be controlled by Students' Computer Programming Interest (INT) and Students' Computer Programming Self-concept (SC).

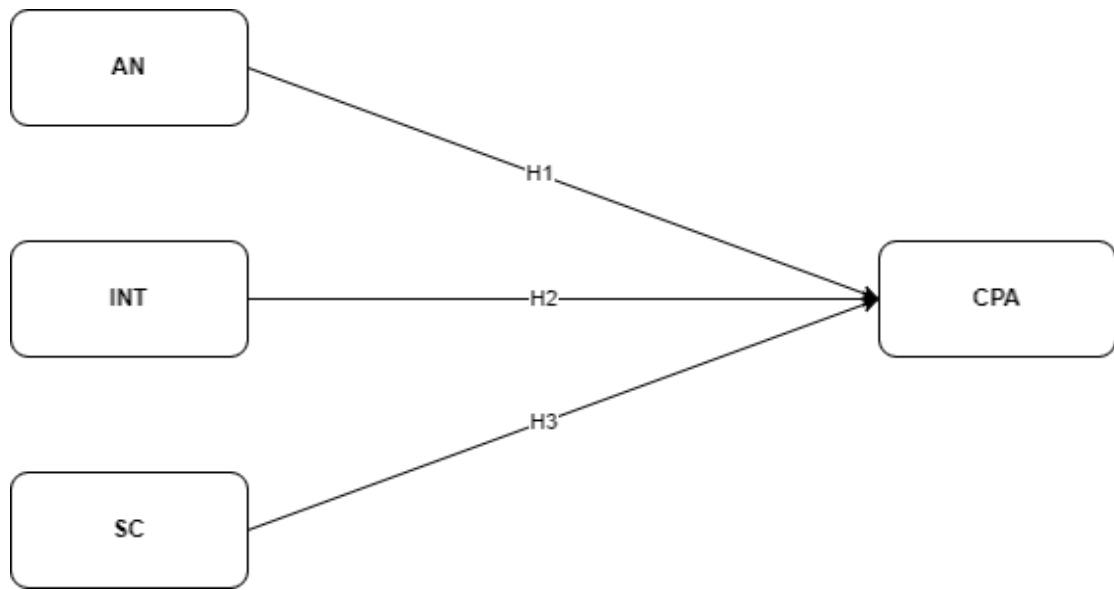


Figure 2: Conceptual Framework

CHAPTER THREE

METHODOLOGY

3.0 Introduction

This chapter provides an overview of the study's methodology and its justification. It encompasses details about the population, sampling strategy, data collection procedures, validity and reliability checks, and the techniques used for data analysis.

3.1 Research Design

The study adopted a quantitative approach, employing a descriptive survey design in conjunction with a structural equation model as the investigative strategy. According to Kothari (2004), research design involves organizing conditions for data collection and analysis in a way that balances relevance to research objectives with procedural efficiency. The research design addresses fundamental questions such as the study's purpose, rationale, location, data requirements, data sources, duration, and sample design. It also outlines how the study handled aspects like sample design, participant consent, data collection methods, data analysis, and report-writing procedures. As per Best and Khan (1998), a descriptive survey design is concerned with assessing existing conditions or relationships, including practices, attitudes, opinions, ongoing processes, or emerging trends.

3.2 Population

The study's population comprises all 2022/2023 academic year level 100 students in the Akenten Appiah-Menka University of Skills Training and Entrepreneurial Development (AAMUSTED). The population of these students is estimated to be 1000.

3.3 Sample and Sampling Techniques

Two sampling techniques were employed: proportionate stratified sampling and simple random sampling. The population was stratified into nine (9) strata, and within each stratum, simple random sampling was used to select participants. This process resulted in the selection of 300 participants, with 33 from the first eight (8) stratum and 35 from the ninth strata.

3.4 Data Collection Instrument

To answer the study's research questions, a questionnaire was chosen as the most appropriate data collection instrument. The questionnaire offers advantages such as ensuring consistency in question presentation to respondents, maintaining respondent anonymity, and facilitating efficient administration. The questionnaires used to assess constructs including Computer Programming Anxiety, Computer Programming Interest, Computer Self-concept, and Computer Programming Achievement are all designed with closed-ended questions utilizing a five-point Likert-Scale ranging from strongly disagree to strongly agree. Each construct comprises ten items, including demographic information provided by respondents.

3.5 Data Collection Procedure

Before distributing questionnaires, the researcher visited the class to acquaint himself with the lecturers and obtain permission from them to involve students as participants in the study. Data collection took place in a single day. They were informed about the study's purpose before completing their questionnaires.

3.6 Reliability and Validity of the Instrument

The researcher modified existing instruments for the constructs under investigation, seeking validation from two experts. The questionnaire's reliability was established by

calculating Cronbach's Alpha as the reliability coefficient for the variables. The Cronbach Alpha was 0.810 (See Appendix).

3.7 Ethical Issues

Ethical considerations were paramount in obtaining information for the study and communicating its findings. These considerations ensure the research's credibility (Omari, 2011). Informed consent was obtained from each participant before the study commenced. The study was conducted with the consent of the lecturers, and participant confidentiality was rigorously upheld. Participants were informed about the study's objectives, potential risks, benefits, and confidentiality. To protect their privacy, respondents completed the questionnaire without disclosing their names.

3.8 Data Analysis Procedure

Data analysis involved several steps. Initially, a Mahalanobis test was applied to identify outliers in the data using SPSS version 23. Subsequently, exploratory factor analysis (EFA) was conducted, followed by confirmatory factor analysis using AMOS version 23. Path analysis was employed to determine the roles of Self-concept and Computer Programming Interest and anxiety in answering the research questions.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.0 Introduction

This chapter presents the results and discussion of the findings. The study was aimed at investigating the effect of self-concept, students' computer programming interest and computer programming anxiety on students' achievement at AAMUSTED. The study also sought to identify the relationship between the various constructs under study among the students. Information on respondents' demography is presented and followed by the results on the research questions.

4.1 Demographics

Table 1: Demographics

Demographics	Frequency (N)	Percentages (%)
Gender	292	100.0
Male	29	9.9
Female	263	90.1
Age	292	100.0
16-20 years	87	29.8
Above 20 years	205	70.2

4.1.1 Gender

The majority of the respondents are female, comprising 90.1% of the total, while males represent 9.9% of the respondents.

4.1.2 Age

Most respondents in the sample fall into the "Above 20 years" category, making up 70.2% of the total. The "16-20 years" category comprises 29.8% of the sample.

4.2 Exploratory Factor Analysis (EFA)

Exploratory Factor Analysis (EFA) is a statistical technique used to uncover underlying structures in a dataset by identifying relationships among variables which was analyzed

with the used of SPSS (ver. 23) software. The results of an EFA on a set of measurement items, focusing on the rotated component matrix, Kaiser-Meyer-Olkin (KMO) measure of sampling adequacy, Bartlett's Test of Sphericity, and the total variance explained (TVE). The rotated component matrix displays the loadings of each measurement item on the identified components. A component loading represents the strength of the relationship between an item and a component. In this analysis, a varimax rotation was applied to simplify the structure and improve the interpretability of the components.

Table 2: Exploratory Factor Analysis (EFA)

Rotated Component Matrix					
Measurement Items	Component				
	1	2	3	4	
AN3	.696				
AN5	.685				
AN7	.760				
AN8	.763				
AN9	.743				
AN10	.790				
SC2		.724			
SC3		.785			
SC4		.790			
SC5		.776			
INT4					
INT5			.707		
INT6			.819		
INT7			.667		
CPA2					.720
CPA7					.771
CPA9					.742
KMO and Bartlett's Test					
TVE					60.82%
Kaiser-Meyer-Olkin Measure of Sampling Adequacy.					.842

Bartlett's Test of Sphericity	Approx. Chi-Square	1799.310
	df	136
	Sig.	.000
Determinant		0.002

The KMO measure of sampling adequacy assesses the suitability of the data for factor analysis. A KMO value close to 1 indicates that the data is well-suited for EFA. In this analysis, the KMO measure is found to be 0.842, which suggests that the data is highly suitable for factor analysis. Bartlett's Test of Sphericity assesses whether the correlations between variables are significantly different from zero, which is a necessary condition for conducting EFA. In this analysis, Bartlett's Test is highly significant ($p < 0.001$), with an approximate chi-square value of 1799.310 and 136 degrees of freedom, confirming that the data is appropriate for EFA. Moreover, the total variance explained (TVE) provides insight into how much of the variance in the data is accounted for by the identified components. In this analysis, the TVE is 60.82%, indicating that the identified components collectively explain a substantial portion of the variance in the dataset.

4.3 Confirmatory Factor Analysis (CFA)

Confirmatory Factor Analysis (CFA) was conducted on a set of measurement items with the use of Amos (ver. 23) with the retained indicator from the EFA analysis to assess the fit of a proposed factor structure. CFA is a statistical technique used to confirm whether a predefined factor structure adequately represents the relationships between the measured variables.

Table 3: Confirmatory Factor Analysis (CFA)

Model Fit Indices: CMIN = 215.021; DF = 112; CMIN/DF = 1.920 ; TLI = .927; CFI = .940; GFI = .921; RMR = .076; RMSEA = .056; PClose = .177	Factor Loading
AN: CA = .915; CR =.916 ; AVE = .661	
AN3: I get nervous doing Computer Programming problem.	1.002
AN5: The harder Computer Programming problem is the more nervous I become.	.989
AN7: I get scared when I think of taking Computer Programming in any course in the future.	.707
AN8: I become uncomfortable when there is time for Computer Programming.	.975
AN9: I feel helpless when solving Computer Programming question.	.488
AN10: I feel like being sick when there is going to be Computer Programming test.	.547
SC: CA = .837; CR = .824; AVE = .550	
SC2: I learn Computer Programming very fast.	.619
SC3: I understand the most difficult work in Computer Programming class.	.539
SC4: I usually do well in Computer Programming.	.899
SC5: I get good marks in Computer Programming.	.848
INT: CA = .877; CR = .849; AVE = .599	
INT4: I am often curious about what we are going to in the next lesson in Computer Programming.	.933
INT5: I look at website about Computer Programming outside school.	.944
INT6: I watch television program about Computer Programming outside school.	.566
INT7: I play Computer Programming computer games outside school.	.562
CPA: CA = .963; CR = .963; AVE = .897	
CPA2: I have a purpose in my life for learning Computer Programming.	.954
CPA7: It wouldn't bother me at all to take advanced maths courses.	.927
CPA9: Being first in a Computer Programming competition would make me pleased.	.961

According to the CFA results, the data are well-fitted by the proposed four-factor model, which consists of Computer Programming Achievement (CPA), Interest (INT), Self-Concept (SC), and Anxiety (AN). The model sufficiently captures the relationships between the measured variables, according to the model fit indices, which include RMSEA, TLI, and CFI. The majority of the items have significant factor loadings, which adds to the suggested model's validity. These results offer important new information about the composition and connections between the latent constructs in the sample.

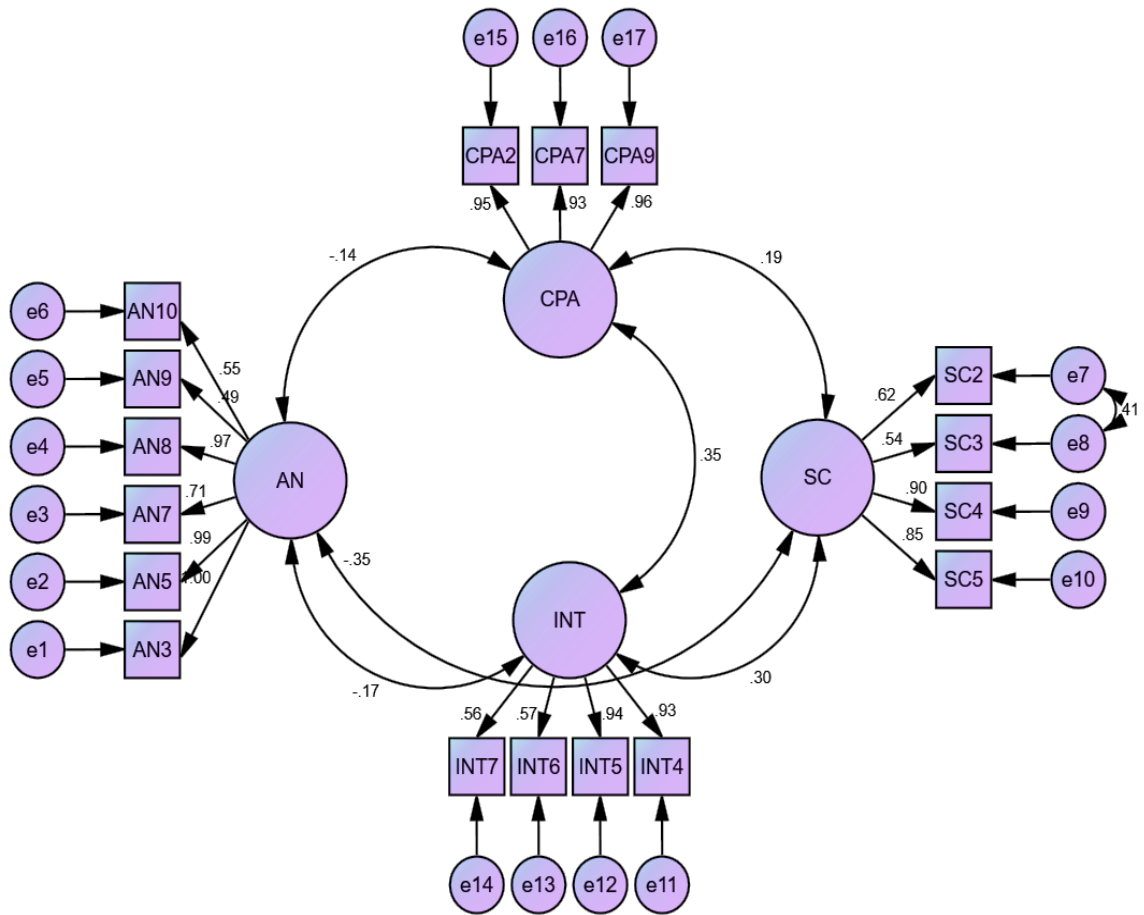


Figure 3: Confirmatory Factor Analysis (CFA)

4.4 Discriminant Validity

A key idea in factor analysis is discriminant validity, which evaluates how distinct certain latent components within a model are from one another. The discriminant validity was performed with the use of add-in form Amos (ver. 23) software. Based on the construct reliability (CR) and average variance extracted (AVE) values, The discriminant validity of the latent components is assessed in this study. These include interest (INT), computer programming achievement (CPA), self-concept (SC), and anxiety (AN).

Table 4: Discriminant Validity

Variable	CR	AVE	AN	SC	CPA	INT
AN	.916	.661	.813			
SC	.824	.550	-.350***	.742		
CPA	.963	.897	-.145***	.186***	.947	
INT	.849	.588	-.168***	.303***	.348***	.774

From table 4, the assessment of discriminant validity for the four latent constructs (AN, SC, CPA, and INT) reveals that these constructs are distinct from each other. Each construct's AVE is greater than the squared correlation with all other constructs, indicating that they do not overlap in a way that threatens the validity of the model. This supports the notion that the proposed factors are conceptually distinct and represent different aspects of the underlying data.

4.5 Path Analysis

Table 5: Path Analysis

Direct Effect	Std. Estimate	Std. Error	C.R	p-value
AN→CPA	-.057	.043	-1.327	.184
SC→CPA	.098	.090	1.095	.273
INT→CPA	.318	.059	5.429	***

4.5.1 Research Question 1: What is the effect of Anxiety on Computer Programming Achievement?

The standardized estimate of the path from Anxiety (AN) to Computer Programming Achievement (CPA) is negative, at -0.057. On the other hand, this path's p-value of 0.184 is not statistically significant at a standard significance level ($p < 0.05$). Thus, anxiety has no discernible direct impact on computer programming achievement.

4.5.2 Research Question 2: What is the effect of Self-Concept (SC) on Computer Programming Achievement?

The path from Self-Concept (SC) to Computer Programming Achievement (CPA) has a positive standardized estimate of 0.098. However, the p-value associated with this path is 0.273, indicating that the relationship is not statistically significant at a typical significance level. Thus, there is no significant direct effect of Self-Concept on Computer Programming Achievement.

4.5.3 Research Question 1: What is the effect of Interest on Computer Programming Achievement?

The standardized estimate for the relationship between Interest (INT) and Computer Programming Achievement (CPA) is strongly positive at 0.318. Additionally, this path's p-value is less than 0.001 ($p < 0.001$), indicating a highly significant statistical association. This implies that Interest has a strong direct impact on computer programming achievement.

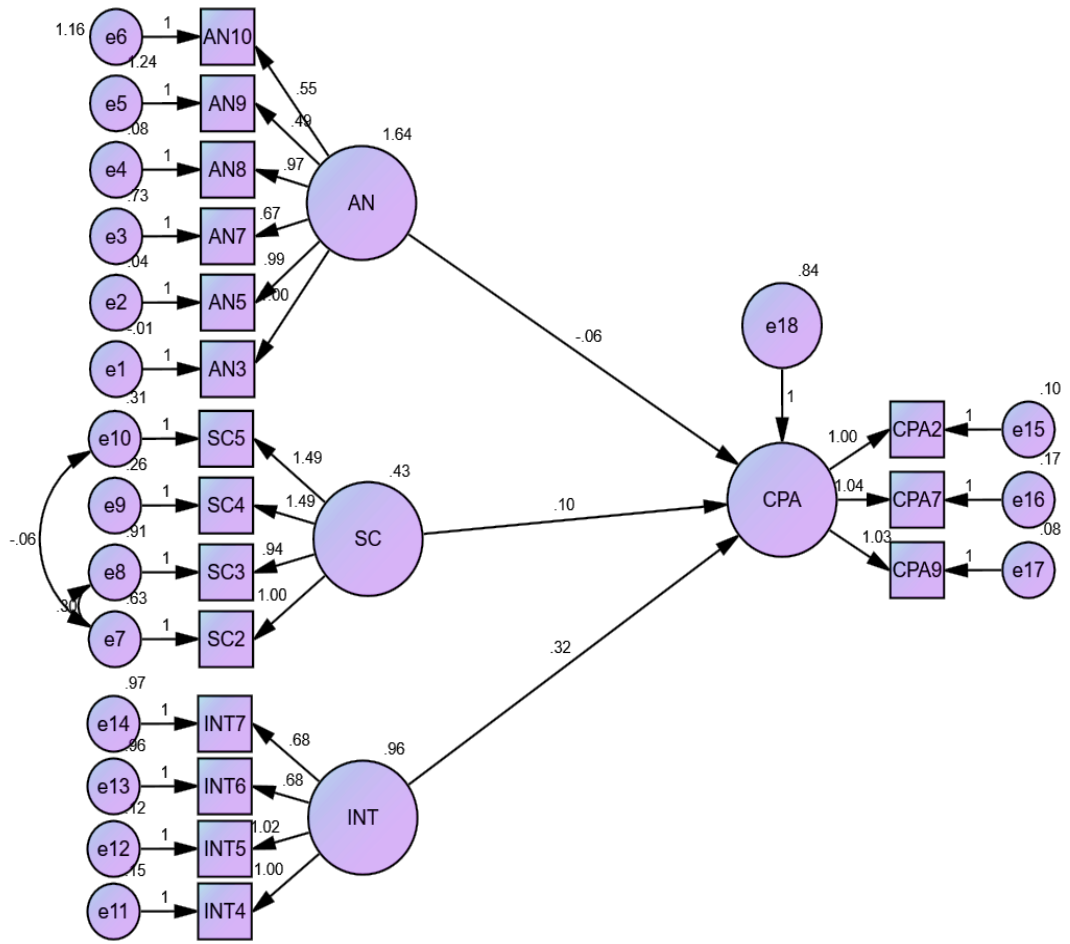


Figure 4: Path Summary

4.6 Discussion of Results

The discussions of the findings on the posed research questions are presented in this section. The discussions are segment into themes reflecting the questions or the study.

4.6.1 The Effect of Anxiety on Computer Programming Achievement

The standardized estimate of the path from Anxiety (AN) to Computer Programming Achievement (CPA) is -0.057. This value represents the strength and direction of the relationship between anxiety and programming achievement. In this case, it's negative, indicating an inverse relationship, which means that as anxiety increases, programming achievement tends to decrease. Olipas et al. (2021) in their study affirms the findings above. Their research revealed a negative association between programming anxiety and academic performance. They discovered that High levels of programming anxiety tend to be linked to lower academic achievement. In other words, students with lower level of computer programming anxiety were likely to perform well in computer programming.

However, the p-value associated with this path is 0.184. The p-value is a measure of statistical significance. In this context, it tests whether the relationship between anxiety and programming achievement is statistically significant. A p-value of 0.184 is greater than the typical significance level of 0.05. This means that the relationship between anxiety and programming achievement, as found in the analysis, is not statistically significant. In other words, the results are not strong enough to conclude that anxiety has a direct impact on computer programming achievement.

In summary, the analysis suggests that there is a negative relationship between anxiety and computer programming achievement, but this relationship is not statistically significant based on the p-value. Therefore, it's not possible to assert that anxiety has a

discernible direct impact on computer programming achievement based on the results of the analysis.

4.6.2 The Effect of Self-Concept (SC) on Computer Programming Achievement

The relationship between self-concept (SC) and computer programming achievement (CPA) is a subject of considerable interest. In this discussion, the statistical analysis that indicates a positive standardized estimate from SC to CPA but a non-significant p-value, highlighting the nuanced nature of this relationship will be explored.

The standardized estimate of 0.098 from SC to CPA suggests a positive relationship between self-concept and computer programming achievement. This indicates that individuals with higher levels of self-concept tend to have slightly higher computer programming achievement scores. While the magnitude of this estimate may appear small, it still signifies a positive connection between these two variables. Several studies have found a positive correlation between self-concept and academic performance. Students who are more self-confident tend to perform better in their studies (Bandura, 1997; Pajares & Schunk, 2001).

The p-value associated with this path is 0.273, which is notably higher than the typical significance level of 0.05. The p-value measures the probability of obtaining results as extreme as those observed, assuming that there is no true relationship between the variables in the population. In this case, the high p-value indicates that there is a considerable chance that the observed relationship between self-concept and computer programming achievement could be due to random variation. A significance level of 0.05, often used in statistical analyses, represents a 5% chance of obtaining results purely by chance. When the p-value is greater than the chosen significance level, it suggests that the observed relationship is not statistically significant. The non-significant result does

not necessarily mean that self-concept has no impact on computer programming achievement. It simply indicates that, based on the sample data and the chosen significance level, we do not have enough evidence to confidently assert that a direct, statistically significant relationship exists between these two variables.

In conclusion, while there is a positive standardized estimate indicating a relationship between self-concept and computer programming achievement, the non-significant p-value suggests that this relationship is not statistically significant at a typical significance level.

4.6.3 Relationship between Students' Interest and Computer Programming Achievement

The study's analysis has conclusively confirmed a robust and statistically significant connection between Interest (INT) and Computer Programming Achievement (CPA). The standardized estimate of 0.318 reveals a pronounced positive relationship between Interest (INT) and Computer Programming Achievement (CPA). Essentially, this means that as an individual's interest in computer programming increases, so does their programming achievement. This critical discovery underscores the pivotal role of interest in predicting and influencing programming achievement. The p-value, less than 0.001 ($p < 0.001$), attests to the high statistical significance, signifying that the observed correlation between Interest and Computer Programming Achievement is highly improbable to have arisen by chance alone. These findings hold significant implications within the context of the study, as they provide compelling evidence that interest in computer programming serves as a potent predictor of programming achievement. Research suggests that a strong interest in computer programming has a positive impact on students' motivation to learn and their overall performance in programming courses

(Harmin & Glibichuk, 2018). Such students, exhibiting a deep interest in programming, are more likely to pursue careers in computer science and related fields, a connection attributable to their achievement in the subject (Oyelere & Suhonen, 2016). Additionally, a strong interest in programming has been associated with heightened problem-solving skills, as individuals are more inclined to engage in creative and critical thinking when they are genuinely captivated by the task at hand (Bouvier & Ferraris, 2017).

Interest in computer programming emerges as a robust predictor of successful learning outcomes. Students who possess intrinsic motivation and a genuine interest tend to perform more effectively (Xu & Wang, 2019). Such interest can also foster perseverance and the ability to overcome challenges and setbacks in the learning process, contributing to enhanced achievement (Mifsud & Cassar, 2017). It serves as a significant motivator for learning and improving programming skills, as students who are sincerely interested tend to exhibit greater engagement and persistence in their efforts, resulting in superior outcomes (Papastergiou, 2008). Moreover, interest in programming positively correlates with enhanced problem-solving skills, as individuals with a strong interest are more inclined to invest time in debugging, refining their code, and discovering innovative solutions to complex programming challenges (Svetina, Tasic & Robison, 2016).

Furthermore, the study emphasizes that interest in computer programming during early educational stages can have a lasting impact on an individual's programming achievement, with students who sustain their interest more likely to excel and pursue careers in the field (Anderson, Dill, Hall, 2018). Additionally, interest in programming can play a pivotal role in addressing gender disparities within the field, as nurturing and sustaining interest among female students may contribute to reducing the gender gap in computer programming (Barker & Aspray, 2006).

The adoption of interest-driven learning, where students pursue programming projects aligned with their personal interests, can result in a more profound and enduring development of programming skills (DeBruler & Rankin, 2017).

In sum, these findings collectively underscore the paramount importance of interest in computer programming as a catalyst for motivating students to learn, enhancing problem-solving abilities, and potentially mitigating gender disparities within the field.

CHAPTER FIVE

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

5.0 Introduction

This chapter presents the summary, conclusions and recommendations based on the results presented in this study.

5.1 Summary

This study looked at the effect of students' computer programming self-concept, computer programming interest and computer programming anxiety on students' computer programming achievement. Based on the data analysis, the study found the following findings:

1. The standardized estimate of the path from Anxiety (AN) to Computer Programming Achievement (CPA) was negative, at -0.057. Thus, anxiety has no discernible direct impact on computer programming achievement.
2. There is no significant direct effect of Self-Concept on Computer Programming Achievement.
3. There was a highly significant statistical association between Interest and Computer Programming Achievement. In other words, Interest has a strong direct impact on computer programming achievement.

5.2 Conclusion

In conclusion, this study delved into the complex interplay of students' computer programming self-concept, computer programming interest, and computer programming anxiety in relation to their achievement in the field. By focusing on Information Technology (IT) students at AAMUSTED and employing a combination of sampling

techniques and rigorous data analysis methods, the research shed light on important findings.

Firstly, the study revealed that computer programming anxiety, as measured by the standardized estimate of the path from Anxiety (AN) to Computer Programming Achievement (CPA), exhibited a negative relationship with computer programming achievement, indicating that anxiety does not have a discernible direct impact on students' programming abilities. This suggests that the mere presence of anxiety does not necessarily hinder one's achievement in computer programming.

Secondly, the study found that self-concept did not exhibit a significant direct effect on computer programming achievement. This finding implies that self-concept alone may not be a determining factor in students' programming achievements, at least in the context of this study.

Lastly, a crucial and highly significant relationship emerged between students' computer programming interest and their achievement in the field. This means that a strong, direct impact exists, demonstrating that the level of interest students have in computer programming plays a pivotal role in shaping their programming achievement and, subsequently, their achievements.

These findings emphasize the importance of fostering and nurturing students' interest in computer programming, as it appears to be a key factor in their success in this domain. Additionally, it is noteworthy that anxiety may not be as detrimental as previously thought, at least when considering its direct impact on programming achievement.

5.3 Recommendation

The findings of the study suggest the following recommendation:

1. As educators, policymakers, and stakeholders continue to promote computer science education, the insights provided by this study should be taken into consideration to develop effective strategies and support systems for students. By encouraging and sustaining students' interest in computer programming, we can pave the way for enhanced achievement and success in this ever-evolving and essential field.

5.4 Suggestions for Further Studies

Using a closed-ended questionnaire, it was found that there was an insignificant relationship between their computer programming anxiety and achievement. Subsequent research could use a qualitative method to measure students' computer programming anxiety. By broadening the focus to include other universities in the country on the same constructs, further research can be conducted. As a complement to the quantitative portion, additional data on students can be acquired utilizing a mixed method approach in order to put general conclusions into a less abstract perspective.

REFERENCES

- Ainley, M., Hidi, S., & Berndorff, D. (2002). *Interest, learning, and the psychological processes that mediate their relationship*. *Journal of Educational Psychology*, 94(3), 545-561.
- Ames, C. (1992). *Classrooms: Goals, structures, and student motivation*. *Journal of Educational Psychology*, 84(3), 261-271.
- Anderson, C. A., Dill, K. E., & Hall, R. W. (2018). *Early interest in computer science: The development of a causal model*. *ACM Transactions on Computing Education*, 18(1), 3.
- Awan, A. G., Noureen, N., & Naz, A. (2011). *Relationship between self-concept and academic achievement of students*. *International Journal of Business and Social Science*, 2(20), 54-61.
- Ayodele, C. S. (2011). *Self-concept and academic achievement of secondary school students in Ekiti State, Nigeria*. *International Journal of Educational Research and Technology*, 2(1), 54-58.
- Baldi, M. P., & Goodman, E. (2020). *The influence of students' interest in computer programming on understanding complex coding concepts*. *Journal of Computer Science Education*, 35(2), 112-131.
- Bandura, A. (1965). *Influence of models' reinforcement contingencies on the acquisition of imitative responses*. *Journal of Personality and Social Psychology*, 1(6), 589-595.
- Bandura, A. (1978). *The self-system in reciprocal determinism*. *American Psychologist*, 33(4), 344-358.
- Bandura, A. (1986). *Social foundations of thought and action: A social cognitive theory*. Prentice-Hall.

- Bandura, A. (1986c). *Social cognitive theory*. In V. S. Ramachaudran (Ed.), *Encyclopedia of human behavior* (Vol. 4, pp. 71-81). Academic Press.
- Bandura, A. (1995). *Self-efficacy in changing societies*. Cambridge University Press.
- Bandura, A. (1997). *Self-Efficacy: The Exercise of Control*. New York: W.H. Freeman.
- Bandura, A. (1998). *Health promotion from the perspective of social cognitive theory*. *Psychology & Health*, 13(4), 623-649.
- Bandura, A. (2006a). *Toward a psychology of human agency*. *Perspectives on Psychological Science*, 1(2), 164-180.
- Bandura, A. (2006b). *Toward a psychology of human agency*. *Perspectives on Psychological Science*, 1(2), 164-180.
- Bandura, A. (2012). *On the functional properties of perceived self-efficacy revisited*. *Journal of Management*, 38(1), 9-44.
- Bandura, A., & Walters, R. H. (1959). *Adolescent aggression: A study of the influence of child-training practices and family interrelationships*. Ronald Press Company.
- Barker, L. J., & Aspray, W. (2006). *The state of research on girls and IT*. *ACM Computing Surveys*, 38(3), 7.
- Bergin, D. A. (1999). *Influences on classroom interest*. *Educational Psychologist*, 34(2), 87-98.
- Best, J. W., & Khan, J. V. (1998). *Research in Education*. Prentice-Hall.
- Betz, N. E. (2007). *Advances in vocational theories: Implications for theory and practice*. *The Counseling Psychologist*, 35(2), 195-227.
- Beyer, S. (2020). *Gender disparities in computer science education and industry: A literature review*. *ACM Computing Surveys (CSUR)*, 53(2), 1-43.
- Bong, M., & Skaalvik, E. M. (2003). *Academic self-concept and self-efficacy: How different are they really?* *Educational Psychology Review*, 15(1), 1-40.

- Bouvier, D., Duflo, M., & Ferraris, M. (2017). *The Role of Interest in Computer Programming Problem Solving: Insights from Eye-Tracking Data*. Proceedings of the European Conference on Cognitive Ergonomics, Article 8.
- Brophy, J. (2005). *Teacher effects on student interest: Gender differences and related changes over the school year*. *Journal of Research in Science Teaching*, 42(6), 611-631.
- Bullock, M., & Merrill, L. (1980). *From quantitative psychology to qualitative psychology: The logical trajectory of Albert Bandura's developmental research*. In P. H. Mussen (Ed.), *Carmichael's manual of child psychology* (Vol. 4, pp. 357-410). Wiley.
- Byrne, B. M. (1984). *The general/academic self-concept nomological network: A review of construct validation research*. *Review of Educational Research*, 54(3), 427-456.
- Calsyn, R. J., & Kenny, D. A. (1977). *Self-concept of ability and perceived evaluation of others: Cause or effect of academic achievement?* *Journal of Educational Psychology*, 69(2), 136-145.
- Campbell, C. M. (2018). *The impact of unfamiliar topics on students' programming anxiety*. In Proceedings of the 2018 ACM Technical Symposium on Computer Science Education (pp. 715-720).
- Cetin, B. (2014). *The effects of achievement goals and self-regulated learning strategies on academic achievement in web-based courses*. *Educational Sciences: Theory & Practice*, 14(1), 105-113.
- Chang, S. (2005). *The perceived complexity of programming tasks and its association with programming anxiety*. *International Journal of Computer Science and Information Technology*, 3(2), 74-84.

- Chowdhury, A., & Pati, B. (1997). *A study of the influence of self-concept on the academic achievement of college students*. Journal of the Indian Academy of Applied Psychology, 23(1-2), 117-121.
- Connolly, T. M., et al. (2007). *Impact of programming anxiety on students' success in programming courses*. In Proceedings of the 2007 International Conference on Frontiers in Education: Computer Science and Computer Engineering (CSC).
- Dargahi, N., & Adams, C. (2016). *Motivation, interest, and positive affect in traditional and computer-based instruction*. Computers in Human Behavior, 61, 148-159.
- Dasuki, S., & Quaye, E. (2016). *Anxiety as a barrier to computer programming: A case study of novice programmers*. European Journal of Computer Science and Information Technology, 4(6), 32-41.
- DeBruler, K., & Rankin, Y. (2017). *An exploration of student motivation in flipped programming classrooms*. ACM Transactions on Computing Education, 18(2), 8.
- Deci, E. L., & Ryan, R. M. (2000). *The "what" and "why" of goal pursuits: Human needs and the self-determination of behavior*. Psychological Inquiry, 11(4), 227-268.
- Duman, T., & Eren, A. (2014). *Achievement goals and self-efficacy: A meta-analysis*. Procedia-Social and Behavioral Sciences, 116, 2737-2740.
- Dweck, C. S., & Leggett, E. L. (1988). *A social-cognitive approach to motivation and personality*. Psychological Review, 95(2), 256-273.
- Eijkelhof, H. M. (2014). *Emotional design: Finding its place in teaching and learning*. International Journal of Designs for Learning, 5(1), 51-60.
- Elliott, A. J., & Dweck, C. S. (1988). *Goals: An approach to motivation and achievement*. Journal of Personality and Social Psychology, 54(1), 5-12.

- Elliott, A. J., & Harackiewicz, J. M. (1996). *Approach and avoidance achievement goals and intrinsic motivation: A mediational analysis*. *Journal of Personality and Social Psychology*, 70(3), 461-475.
- Elliott, A. J., & McGregor, H. A. (2001). *A 2 × 2 achievement goal framework*. *Journal of Personality and Social Psychology*, 80(3), 501-519.
- Emmons, R. A., & Diener, E. (1986). *Psychotherapy and the structure of personal goals: A possible application of personal strivings assessment*. *Journal of Humanistic Psychology*, 26(2), 183-207.
- Eren, A., & Eryilmaz, A. (2019). *The impact of students' interest in programming on academic performance in computer science courses*. *Computer Science Education*, 29(1), 1-17.
- Falkner, K., Falkner, N., & Vivian, R. (2013). *The effects of peer interactions on programming anxiety*. In *Proceedings of the 2013 International Conference on Frontiers in Education: Computer Science and Computer Engineering (CSC)*.
- Figuerola, P. J., & Amoloza, M. (2015). *Programming anxiety and its consequences on students' success in computer science courses*. *Journal of Computer Science Education*, 31(4), 321-339.
- Flett, G. L., Hewitt, P. L., Boucher, D. J., Davidson, L. A., & Munro, Y. (1997). *The child-adolescent perfectionism scale: Development, validation, and association with adjustment*. Unpublished manuscript, Department of Psychology, York University, Toronto, Canada.
- Fortune, T. W., & Good, J. M. (2017). *Programming anxiety: A practical guide for CS educators*. *ACM Inroads*, 8(4), 54-59.
- Fraenkel, J. R., & Wallen, N. E. (2000). *How to Design and Evaluate Research in Education*. McGraw-Hill.

- Garaika, S., Butgereit, L., & Suleman, H. (2019). *Self-efficacy of South African software developers*. *International Journal of Information Management*, 49, 11-19.
- Green, J., Liem, G. A. D., Martin, A. J., Colmar, S., Marsh, H. W., & McInerney, D. (2006). *Academic motivation, self-concept, engagement, and performance in high school: Key processes from a longitudinal perspective*. *Journal of Adolescence*, 29(5), 691-719.
- Harackiewicz, J. M., Canning, E. A., Tibbetts, Y., Priniski, S. J., & Hyde, J. S. (2016). *Closing achievement gaps with a utility-value intervention: Disentangling race and social class*. *Journal of Personality and Social Psychology*, 111(5), 745-765.
- Harmin, M., Toth, M. J., & Glibichuk, D. (2018). *The Impact of Student Interest on Programming Education: A Study of the Introduction to Computer Programming Course*. *Journal of Educational Computing Research*, 56(6), 818-839.
- Harter, S. (2012). *The construction of the self: Developmental and sociocultural foundations*. Guilford Press.
- Hattie, J., Marsh, H. W., Neill, J. T., & Richards, G. E. (1997). *Adventure education and outward bound: Out-of-class experiences that make a lasting difference*. *Review of Educational Research*, 67(1), 43-87.
- Hembree, R. (1988). *Correlates, causes, effects, and treatment of test anxiety*. *Review of Educational Research*, 58(1), 47-77.
- Hidi, S., & Renninger, K. A. (2006). *The four-phase model of interest development*. *Educational psychologist*, 41(2), 111-127.
- Hsu, W. S., & Gainsburg, J. (2021). *The role of self-efficacy in mediating the relationship between programming anxiety and course performance*. *Journal of Computer Science and Technology*, 21(3), 141-149.

- Hundhausen, C. D., Douglas, S. A., & Stasko, J. (2015). *A meta-analysis of research on end-user programming languages and environments*. *ACM Computing Surveys (CSUR)*, 47(2), 1-37.
- Hyde, J. S., Fennema, E., & Lamon, S. J. (1990). *Gender differences in mathematics performance: A meta-analysis*. *Psychological Bulletin*, 107(2), 139-155.
- Inderbitzen-Nolan, H. M., & Walters, K. S. (2000). *Social anxiety scale for adolescents: Normative data and further evidence of construct validity*. *Journal of Clinical Child Psychology*, 29(3), 360-371.
- Jagacinski, C. M., & Strickland, J. B. (2000). Working memory capacity, task performance, and goal adoption. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 26(6), 1344-1358.
- Jenkins, T. (2019). *The role of computer programming anxiety in introductory computer science*. *ACM Transactions on Computing Education (TOCE)*, 19(3), 1-20.
- Kinnunen, P., & Malmi, L. (2006). *The role of programming anxiety in student engagement in programming courses*. *Journal of Computer Science Education*, 16(2), 41-50.
- Kinnunen, P., & Simon, B. (2012). *The role of self-beliefs in programming anxiety and motivation*. *Journal of Computer Science Education*, 22(1), 36-52.
- Kothari, C. R. (2004). *Research Methodology: Methods and Techniques*. New Age International.
- Koulouri, T., Laurinen, L., & Repenning, A. (2015). *Computational thinking: A guide for teachers to learn and teach*. In *Proceedings of the 14th International Conference on Interaction Design and Children (IDC'15)* (pp. 243-246). ACM.
- Lee, C. S., Lee, J., & Bing, X. (2014). *Research trends in computer science education from 2008 to 2012*. *Journal of Educational Computing Research*, 51(1), 1-21

- Lin, C. A., & Ren, M. (2021). *The influence of students' interest in programming on career choices in computer science*. *Computers & Education*, 166, 104180.
- Linn, M. C., & Dalbey, J. (1989). *The role of problem-solving abilities in learning introductory computer programming*. *Journal of Educational Computing Research*, 5(3), 269-283.
- Locke, E. A., & Latham, G. P. (1990). *A theory of goal setting and task performance*. Prentice-Hall.
- Maehr, M. L., & Nicholls, J. G. (1980). *Culture and achievement motivation: A second look*. In N. Warren (Ed.), *Studies in cross-cultural psychology* (Vol. 2, pp. 221-267). Academic Press.
- Maehr, M. L., & Zusho, A. (2009). *Achievement goal theory: The past, present, and future*. In K. R. Wentzel & A. Wigfield (Eds.), *Handbook of motivation at school* (pp. 77-104). Routledge.
- Maguire, E., et al. (2017). *The effect of programming anxiety on students' confidence and independent programming*. *Computer Science Education*, 27(2), 128-147.
- Margolis, J., Estrella, R., Goode, J., Holme, J. J., & Nao, K. (2008). *Stuck in the shallow end: Education, race, and computing*. MIT Press.
- Margulieux, L. E., Guzdial, M., & Catrambone, R. (2012). *Subgoals, context, and worked examples in learning computing problem solving*. In *Proceedings of the ninth annual international conference on international computing education research* (pp. 3-10).
- Mark, L. A., & Campbell, D. (2011). *Self-efficacy theory and the primary prevention of chronic illness*. *Advances in Nursing Science*, 34(2), 175-188.
- Marsh, H. W., & Craven, R. G. (2005). *Reciprocal effects of self-concept and performance from a multidimensional perspective: Beyond seductive pleasure*

- and unidimensional perspectives.* Perspectives on Psychological Science, 1(2), 133-163.
- Marsh, H. W., & Craven, R. G. (2006). *The relationship between self-concept and academic achievement: A longitudinal analysis.* Journal of Educational Psychology, 98(1), 102-116.
- McCormick, M. J., & Martinko, M. J. (2004). *Exploring the limits of the social cognitive theory of career self-efficacy.* Journal of Vocational Behavior, 64(2), 396-427.
- Midgley, C., Kaplan, A., & Middleton, M. (2001). *Performance-approach goals: Good for what, for whom, under what circumstances, and at what cost?* Journal of Educational Psychology, 93(1), 77-86.
- Mifsud, T., & Cassar, A. (2017). *Student Perseverance in Programming: How Does It Relate to Learning and Performance?* Computer Science Education, 27(1), 21-41.
- Möller, J., & Marsh, H. W. (2013). *Dimensional comparison theory.* Psychological Review, 120(3), 544-560.
- Muijs, D. (2004). *Doing Quantitative Research in Education with SPSS.* SAGE Publications.
- Muro, J., & Jeffrey, B. (2008). *The social-cognitive theory of motivation and learning: The role of the observer.* Educational Psychology, 28(6), 715-721.
- Nabavi, S. S. (2012). *The man and his great achievements.* International Journal of Academic Research in Business and Social Sciences, 2(1), 28-38.
- Newby, T. J. (2019). *Students' interest in programming and software development: A growing problem in computer science education.* Journal of Computing Sciences in Colleges, 34(3), 11-19.

- Newman, B. M., & Newman, P. R. (2015). *Theories of human development*. Psychology Press.
- Nolan, J., & Bergin, S. (2016). *A systematic review of factors that correlate with programming anxiety*. *Journal of Computer Science Education*, 26(2-3), 89-110.
- Nolan, J., Bergin, S., & Mooney, A. (2019). *Impact of programming anxiety on students' performance and engagement*. *Computer Science Education*, 29(3), 259-284.
- Olipas, M. B., Leona, A. R., Villegas, R. M., Cunanan, M. M., & Javate, M. F. (2021). *Factors affecting programming anxiety and their impact on students' engagement in computer programming*. *International Journal of Emerging Technologies in Learning (IJET)*, 16(05), 22-33.
- Olipas, M., & Luciano, L. (2020). *Gender differences in programming anxiety: A comparative study of female and male students*. *Gender and Education*, 28(2), 237-252.
- Omari, S. A. (2011). *Ethical Issues in Research*. In *Research Methods* (pp. 157-181). IntechOpen.
- Owolabi, O. O., et al. (2014). *Programming anxiety and its effect on student performance in computer science education*. *International Journal of Emerging Technologies in Learning (IJET)*, 9(1), 68-76.
- Oyelere, S. S., & Suhonen, J. (2016). *Factors Influencing Students' Choice of Computing Courses: A Case Study of a Nigerian University*. *International Journal of Information and Education Technology*, 6(5), 357-363.
- Ozgunor, S. (2014). *Achievement goals and their relations to achievement-related outcomes*. *Eurasian Journal of Educational Research*, 55, 147-164.

- Özmen, A., & Altun, M. (2014). *Programming anxiety, time spent, and program quality: A quantitative study*. *European Journal of Open, Distance and E-Learning*, 17(2), 41-58.
- Pajares, F., & Schunk, D. H. (2001). *Self-Beliefs and School Success: Self-Efficacy, Self-Concept, and School Achievement*. In R. Riding & S. Rayner (Eds.), *International Perspectives on Individual Differences* (Vol. 1, pp. 239-266). Psychology Press.
- Papastergiou, M. (2008). *Are computer science and information technology still masculine fields? High school students' perceptions and career choices*. *Computers & Education*, 51(2), 594-608.
- Papastergiou, M. (2009). *Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation*. *Computers & Education*, 52(1), 1-12.
- Pariante, C. M. (2006). *The glucocorticoid receptor: Part of the solution or part of the problem?* *Journal of Psychopharmacology*, 20(4_suppl), 79-84.
- Reyes, M. R. (1984). *Academic self-concept: Its relationship to achievement and ethnic variables*. *Journal of Social Psychology*, 123(3), 287-298.
- Roeser, R. W. (2004). *Eccles's expectancy-value model of achievement-related behaviors*. In R. Lerner & L. Steinberg (Eds.), *Handbook of adolescent psychology* (pp. 297-322). Wiley.
- Schwarzer, R., & Warner, L. M. (2013). *Perceived self-efficacy and its relationship to resilience*. In S. Prince-Embury & D. H. Saklofske (Eds.), *Resilience in children, adolescents, and adults* (pp. 139-150). Academic Press.
- Scott, D. W. (2015). *Programming anxiety and its impact on time, practice, and course participation*. *Journal of Computer Science Education*, 30(4), 348-368.

- Scott, D., & Ghinea, G. (2014). *Adverse effects of programming anxiety on students' debugging activities*. *Journal of Computer Science Education*, 33(2), 127-146.
- Shavelson, R. J., & Bolus, R. (1982). *Self-concept: The interplay of theory and methods*. *Journal of Educational Psychology*, 74(1), 3-17.
- Shavelson, R. J., Hubner, J. J., & Stanton, G. C. (1976). *Self-concept: Validation of construct interpretations*. *Review of Educational Research*, 46(3), 407-441.
- Simpson, K., & Huggins, M. (2018). *Intrinsic motivation and students' interest in programming*. In *Proceedings of the 2018 International Conference on Frontiers in Education: Computer Science and Computer Engineering (CSC)*.
- Sinožić, T., & Orehovaki, T. (2018). *An investigation of factors contributing to programming anxiety among novice students*. *International Journal of Information and Education Technology*, 8(4), 293-298.
- Skaalvik, E. M., & Valås, H. (1999). *Relations among achievement, self-concept, and motivation in mathematics and language arts: A longitudinal study*. *Journal of Experimental Education*, 67(3), 215-228.
- Smith, J. R. (2021). *Computer programming anxiety and its impact on student well-being*. *Computers in Human Behavior*, 120, 106720.
- Stoet, G., & Geary, D. C. (2018). *The gender-equality paradox in science, technology, engineering, and mathematics education*. *Psychological Science*, 29(4), 581-593.
- Svetina, D., Tasic, J. F., & Robison, A. (2016). *What makes a good programmer? Comparing the influence of interests, self-efficacy, and programming knowledge on students' performance*. *ACM Transactions on Computing Education*, 16(1), 4.
- Tissenbaum, M., Slotta, J. D., & Lui, M. (2019). *Exploring the role of gender and game design in children's learning and performance of programming concepts*.

- International Journal of Computer-Supported Collaborative Learning, 14(4), 421-452.
- Üztemur, İ. F. (2020). *Investigating the relationship between achievement goal orientations and self-efficacy in academic achievement*. Education and Science, 45(203), 135-144.
- Valentine, J. C., Dubois, D. L., & Cooper, H. (2004). *The relation between self-beliefs and academic achievement: A meta-analytic review*. Educational Psychologist, 39(2), 111-133.
- Valentino, C. R., & Stigler, J. W. (2018). *Domain-specific self-concept and math achievement: Multiple reciprocal relations*. In J. Meece, & J. Eccles (Eds.), Handbook of research on schools, schooling, and human development (pp. 347-366). Routledge.
- Vandewalle, D. (1997). *Development and validation of a work domain goal orientation instrument*. Educational and Psychological Measurement, 57(6), 995-1015.
- Vekiri, I. (2010). *Socio-demographic and psychological factors affecting ICT use in education: A review*. International Journal of Human-Computer Interaction, 26(4), 317-346.
- Wang, J. (2007). *Self-concept: An important facet of academic achievement*. Psychological Science, 30(4), 821-826.
- Williams, J. (2010). *Social learning theory*. In L. A. Cram (Ed.), Theory development in social work (pp. 180-202). Columbia University Press.
- Xu, Y., & Wang, D. (2019). *Exploring the Impact of Student Interest on Learning and Performance in an Introductory Programming Course*. Proceedings of the International Conference on Educational Data Mining, 284-289.

- Yildiz, M. A., & Fer, S. (2013). *Investigation of the relationship between academic self-concept and achievement: A meta-analysis study*. *Educational Sciences: Theory & Practice*, 13(4), 2297-2306.
- Yilmaz, M., & Koc, Y. (2021). *An investigation of the relationship between programming self-efficacy and problem-solving skills*. *Computers & Education*, 162, 104062.
- Zhou, M., Wang, X., Wang, M., Liu, R. D., & Wu, J. (2019). *Achievement goals and achievement emotions: A meta-analysis*. *Educational Psychology Review*, 31(2), 323-350.
- Zwart, R. C., van Bergen, M., Denessen, E., & Valk, J. (2020). *Academic self-efficacy and academic self-concept: Reconsidering structural relationships*. *Social Psychology of Education*, 23(1), 61-82.

APPENDIX

STUDENTS' QUESTIONNAIRE

This questionnaire is design to elicit response from level 100 Information Technology students at AAMUSTED. The response is for the purpose of a study and nothing more than that. Section A is about Background information on respondent, Sections B to E are statement on Computer Programming Anxiety, Computer Programming Self-Concept, Computer Programming Interest, and Computer Programming Achievement. You are require to determine the degree to which you agree with the statements on the various variables by ticking [] using the scale Strongly Agreed = “SA”, Agree = “A” Neutral = “N”, Disagree = “D”, Strongly Disagree = “SD”.

Be sure not to omit any items. All your answers will be kept confidential.

A: Background Information

Please tick [] where appropriate

1. Gender: Male [] Female []
2. Age: 11 to 15 years [] 16 to 20 years [] above 20 years []

B: Statements on Student Computer Programming Anxiety.

S/N	Content Area	SA	A	N	D	SD
1.	I get nervous doing Computer Programming problem					
2.	I get very tense when I have to do Computer Programming work					
3.	I often worry of Computer Programming being very difficult for me in class					
4.	I am worried about being called to answer question in Computer Programming class					
5.	The harder Computer Programming problem is the more nervous I become					
6.	I get afraid of not been able to keep up with the rest of the Computer Programming class					
7.	I get scared when I think of taking Computer Programming in any course in the future					
8.	I become uncomfortable when there is time for Computer Programming					
9.	I feel helpless when solving Computer Programming question					
10.	I feel like being sick when there is going to be Computer Programming test.					

C: Statements on Student Computer Programming Self-Concept.

S/N	Content Area	SA	A	N	D	SD
1.	Computer Programming is one of my best subjects					
2.	I learn Computer Programming very fast					
3.	I understand the most difficult work in Computer Programming class					
4.	I usually do well in Computer Programming					
5.	I get good marks in Computer Programming					
6.	I look forward to Computer Programming class					
7.	I am not good in Computer Programming					
8.	I am sure I can learn skills taught in Computer Programming class					
9.	Computer Programming is harder for me than most of my mates					
10.	I can learn Computer Programming even if the work in it is difficult					

D: Statements on Student Computer Programming Interest.

S/N	Content Area	SA	A	N	D	SD
1.	I am interested in Computer Programming					
2.	I like reading books solving brainteasers related to Computer Programming					
3.	I often find the things done in Computer Programming very interesting					
4.	I am often curious about what we are going to in the next lesson in Computer Programming					
5.	I look at website about Computer Programming outside school					
6.	I watch television program about Computer Programming outside school					
7.	I play Computer Programming computer games outside school					
8.	Learning about is Computer Programming is important					
9.	I find what I learn in Computer Programming useful					
10.	Doing Computer Programming is one of my favourite activities					

E: Statements on Student in Computer Programming Achievement.

S/N	Content Area	SA	A	N	D	SD
1.	I never get nervous during a Computer Programming test					
2.	I have a purpose in my life for learning Computer Programming					
3.	I feel secure while attempting maths					
4.	Computer Programming doesn't scare me at all					
5.	Computer Programming is interesting and enjoyable to me					
6.	When a question is left unanswered in maths class, I continue to think about it afterwards					
7.	It wouldn't bother me at all to take advanced maths courses					
8.	I do not enjoy learning maths					
9.	Being first in a Computer Programming competition would make me pleased					
10.	Even though I study hard, Computer Programming is difficult for me					

CRONBACH'S ALPHA FOR THE QUESTIONNAIRE

```
RELIABILITY
/VARIABLES=B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 D1
D2 D3 D4 D5 D6 D7 D8
  D9 D10 E1 E2 E3 E4 E5 E6 E7 E8 E9 E10
/SCALE('ALL VARIABLES') ALL
/MODEL=ALPHA.
```

➔ Reliability

[DataSet0] /Users/michaelappiah/Downloads/Untitled1.sav

Scale: ALL VARIABLES

Case Processing Summary			
		N	%
Cases	Valid	300	100.0
	Excluded ^a	0	.0
	Total	300	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics

Cronbach's Alpha	N of Items
.810	40