

# An Investigation into the Correlation between Programming Self-Efficacy and Debugging Anxiety

Muyaba Angela\*

*\*Dmi St Eugene University.*

## Abstract

This study developed the Programming Self-Efficacy and Debugging Anxiety Questionnaire (PSEDAQ) to assess first-year university students' programming self-efficacy and debugging anxiety. The tool was found to be reliable, valid, and efficient through statistical analysis and student interviews. Results showed that students who performed well in previous programming exams had higher self-efficacy and lower anxiety. No significant differences were observed based on gender or prior programming experience. The questionnaire identified key areas such as general self-efficacy, debugging anxiety, and classroom-related confidence. The study concludes that PSEDAQ is a useful tool for understanding students' learning challenges and improving programming achievement, though further refinement is recommended.

**Keywords:** Programming Self-Efficacy, Debugging Anxiety, Questionnaire Development, Programming Achievement, University Students.

## 1. INTRODUCTION

In Zambia the importance of Programming knowledge is apparent in the employment area as well as in everyday activities. Despite its importance, many people continue to find that the application of Programming knowledge at even basic level difficult (Bruning, Schraw, and Ronning, 1999) and, therefore, they lack the skill to secure employment in Programming - related fields (Rivera Batiz, 1992). This chapter discusses background to the study title Programming self-efficacy and Debugging anxiety questionnaire a case of George Benson University. It also includes statement of the problem, objectives of the study, research questions and significance of the study, limitation, delimitations, abbreviation, the conceptual framework and theoretical framework

### Background of the Problem

As university Programming teachers respond to the need for fostering students' Programming literacy, the important role of students' Programming self-efficacy has received increased attention (Hannula, 2006; Pape & Smith, 2002). Bandura identified four main sources of self-efficacy: mastery experiences, vicarious experiences, social persuasion, and physiological states. Students base most of their beliefs about their abilities on their mastery experiences.

Poor Programming self-efficacy in students often decreases their motivation to learn and eventually can lead to low Programming achievement. In a study of students enrolled in a Programming class, Higbee and Thomas (1999) found that Programming self-efficacy, along with other affective factors such as test Debugging anxiety and perceived usefulness of Programming, influenced students' Programming performances. The results of their study suggest to teachers that focusing on teaching Programming content is insufficient for some students to learn Programming. The impact of Debugging anxiety varies based on each individual student. Students who suffer from higher levels of Debugging anxiety typically develop negative attitudes and emotions toward programming. By the time students participate in future Programming class, their attitudes toward Programming are relatively stable; those

students with debugging anxiety are more likely to avoid taking Programming courses. Perhaps the most severe consequence of Debugging anxiety is a decreased level of Programming achievement.

### **Statement of the Problem**

Because Programming self-efficacy and Debugging anxiety influence students' Programming achievement, it is important to understand how self-efficacy and debugging anxiety relate to each other. Previous research has focused on measuring and exploring the two constructs separately. Because of the possible interrelationship between these two constructs, it would be beneficial to examine them together to answer questions such as the following. (1) Do students with high levels of Programming self-efficacy have low levels of Debugging anxiety? (2) If teachers reduce their students' Debugging anxiety, will the students' self-efficacy in Programming increase? A strong relationship between Programming self-efficacy and Debugging anxiety could have implications for how researchers understand and measure these constructs and how teachers attempt to improve students' attitudes toward Programming.

### **Purpose of the Study**

This research is guided by the following general and specific objectives;

#### **General Objectives**

To develop a questionnaire that can be used to explore the correlation between Programming self-efficacy and debugging anxiety in university students of George Benson.

#### **Specific Objectives**

- To investigate how Programming self-efficacy and Debugging anxiety is related to students' gender,
- To investigate how Programming self-efficacy and debugging anxiety is related to students' secondary school Programming preparation.
- To investigate how Programming self-efficacy and Debugging anxiety is related to students' high school Programming experiences
- To investigate if teachers reduce their students' debugging anxiety.

#### **Research Questions**

- How is Programming self-efficacy and Debugging anxiety related to each other?
- How is Programming self-efficacy and debugging anxiety related to a student's gender?
- How is Programming self-efficacy and debugging anxiety related to students' secondary school Programming preparation?
- How is Programming self-efficacy and Debugging anxiety related to students' high school Programming experiences?
- If teachers reduce their students' Debugging anxiety, will their self-efficacy in Programming increase?

#### **Significance of the Study**

The overall goal of this study is to develop a questionnaire that can be used to explore the relationship between Programming self-efficacy and Debugging anxiety.

The questionnaires that are currently used in research on Programming self-efficacy or Debugging anxiety were designed to be used as separate instruments for a variety of different purposes

In order to investigate the relationship between Programming self-efficacy and Debugging anxiety, researchers need a questionnaire especially designed to explore how these constructs relate to each other. The questionnaire can also be used to explore the relationship of self-efficacy and anxiety with other variables, such as gender, achievement, and prior coursework.

### **Limitation of the Study**

These are challenges that the research might encounter during the collection of data, these limitations are based on assumptions, because the researcher may and may not encounter them at all. Therefore the limitation to this study includes;

- The researcher might anticipate shortage of sample to conduct the study on.
- Respondents might produce bias and prejudices responses.
- The researcher anticipates shortage of transport due to distance.

### **Delimitation of the Study**

These are the boundaries or extent to which the research is limited, therefore the delimitations of this study includes;

- This research is delimited to Programming self-efficacy and Debugging anxiety questionnaire.
- The research is delimited to First year students, teachers and Administrators of George Benson University only.

### **Hypothesis (Basic Assumption)**

The following assumption will guide this study.

- During this research, it is assumed that the correlation will be positive
- During this research, it is assumed that respondents will be cooperative and willing to give information correctly and to the fullest.
- Data collection and analysis are assumed to be carried out as planned.
- Respondents will complete the instruments honestly and objectively.

### **Conceptual Framework**

The conceptual framework of this research proposal is shown below

#### **Dependent variables**

- Programming self-efficacy and Debugging anxiety

#### **Independent variables**

- Gender
- Secondary school Preparation
- High school Experience
- Teachers Attitude

## **2. THEORITICAL FRAMEWORK**

### **Social Cognitive Theory**

Psychologist Albert Bandura has defined self-efficacy as one's belief in one's ability to succeed in specific situations or accomplish a task. One's sense of self-efficacy can play a major role in how one approaches goals, tasks, and challenges. The theory of self-efficacy lies at the center of Bandura's social cognitive theory, which emphasizes the role of observational learning and

social experience in the development of personality. The main concept in social cognitive theory is that an individual's actions and reactions, including social behaviors and cognitive processes, in almost every situation are influenced by the actions that individual has observed in others. Because self-efficacy is developed from external experiences and self-perception and is influential in determining the outcome of many events, it is an important aspect of social cognitive theory. Self-efficacy represents the personal perception of external social factors. According to Bandura's theory, people with high self-efficacy, that is, those who believe they can perform well are more likely to view difficult tasks as something to be mastered rather than something to be avoided

### **Social Learning Theory**

Social learning theory describes the acquisition of skills that are developed exclusively or primarily within a social group. Social learning depends on how individuals either succeed or fail at dynamic interactions within groups, and promotes the development of individual emotional and practical skills as well as accurate perception of self and acceptance of others. According to this theory, people learn from one another through observation, imitation, and modeling. Self-efficacy reflects an individual understands of what skills he/she can offer in a group setting.

### **Self-Concept Theory**

Self-concept theory seeks to explain how people perceive and interpret their own existence from clues they receive from external sources, focusing on how these impressions are organized and how they are active throughout life. Successes and failures are closely related to the ways in which people have learned to view themselves and their relationships with others. This theory describes self-concept as learned (i.e., not present at birth); organized (in the way it is applied to the self); and dynamic (i.e., ever-changing, and not fixed at a certain age).

### **Relationship of theory to the research title**

Students generally avoid tasks where self-efficacy is low, but undertake tasks where self-efficacy is high. When self-efficacy is significantly beyond actual ability, it leads to an overestimation of the ability to complete tasks. On the other hand, when self-efficacy is significantly lower than actual ability, it discourages growth and skill development. Research shows that the optimum level of self-efficacy is slightly above ability; in this situation, people are most encouraged to tackle challenging tasks and gain experience

### **Defination of key terms**

**Programming Self-Efficacy**; refers to an individual's confidence about completing a variety of tasks, from understanding concepts to solving problems, in Programming.

**Debugging anxiety**; refers to a person feeling tense or anxious when working with code or solving Programming problems

### **Abbreviations**

PSES-Programming Self-Efficacy Scale

PSES-R -Programming Self-Efficacy Scale-Revised

PDARS -Programming Debugging anxiety Rating Scale

PSEDAQ -Programming Self-Efficacy and debugging anxiety Questionnaire

## **3. LITERATURE REVIEW**

To understand what types of items are appropriate for a questionnaire regarding first year students' Programming self-efficacy and debugging anxiety, it is essential to understand how researchers define these constructs and what is currently known about them.

### **Programming Self-Efficacy**

Programming self-efficacy is defined as an individual's beliefs or perceptions with respect to his or her abilities in Programming (Bandura, 1997). In other words, an individual's Programming self-efficacy is his or her confidence about completing a variety of tasks, from understanding concepts to solving problems, in Programming. Self-efficacy, in general, has been linked with motivation. It has been well established that students with higher levels of self-efficacy tend to be more motivated to learn than their peers and are more likely to persist when presented with challenges (Pajares & Graham, 1999; Pajares & Kranzler, 1995; Zeldin, Britner & Pajares, 2008).

Although the development of self-efficacy is not fully understood, researchers have consistently confirmed Bandura's (1997) four main sources of self-efficacy: mastery experiences, vicarious experiences, social persuasion, and physiological states (Hampton & Mason, 2003; Lopez & Lent, 1992; Usher & Pajares, 2009). In a study on designing a scale to explore the sources of Programming self-efficacy, Usher and Pajares (2009) found that "perceived mastery experience is a powerful source of students' Programming self-efficacy. Students who feel they have mastered skills and succeeded at challenging assignments experience a boost in their efficacy beliefs"

### **Measuring Programming Self-Efficacy**

The most commonly used scale for measuring Programming self-efficacy is the Programming Self-Efficacy Scale (PSES) (Betz & Hackett, 1983). This scale was originally developed to explore gender differences in Programming self-efficacy and how these differences affect students' career choices. After reviewing previous research on Debugging anxiety and Programming self-efficacy, Betz and Hackett identified three main domains involved with studying Programming self-efficacy: solving Programming problems, using Programming in everyday tasks, and obtaining good grades in Programming. The PSES asks participants to rate their confidence on a scale from 0 to 9 in their ability to perform 18 Programming tasks, to correctly solve 18 Programming problems, and to get a 70% or better in 16 Programming-related subjects. Although no factor analytic research has been conducted on the original PSES, Kranzler and Pajares (1997) used factor analytic techniques to analyze a revised version of the PSES, referred to as the Programming Self-Efficacy Scale-Revised (PSES-R) (Pajares & Miller, 1995). The items on the PSES-R were taken from the original PSES, but the Programming problems were replaced by problems from arithmetic, algebra, and geometry taken from the Programming Confidence Scale (Dowling, 1978). Also, on the PSES-R, students rated their confidence on a scale from 1 to 5, not 0 to 9 as in the original PSES. Factor analysis revealed three factors of the PSES-R, as expected: Programming problems, Programming tasks, and Programming. The subjects however, were split into two factors; pure Programming and science that require a lot of Programming. The identification of multiple factors of the PSES-R suggests that Programming self-efficacy is conceptually more complex than Betz and Hackett (1983) believed.

Although a score can be computed for the PSES, Kranzler and Pajares (1997) cautioned researchers that it is difficult to assign and make appropriate use of an overall score for Programming self-efficacy based on scales such as the PSES or PSES-R. It is important for researchers and educators to consider the multiple factors involved when assessing a student's

level of Programming self-efficacy. Because of the nature of Programming self-efficacy, students can have, or lack, confidence in a multitude of areas involved with Programming. If a student's score is lower on one factor than the rest of the factor scores on a Programming self-efficacy scale, his or her overall score can be distorted, which can lead educators or researchers to misjudge the student's overall level of Programming self-efficacy.

Many of the initial research studies conducted on students' Programming self-efficacy sought to explore how students' Programming self-efficacy influenced their career choices. Betz and Hackett (1983) developed the PSES, as discussed previously, specifically to determine how Programming self-efficacy and gender influence students' choices of science-based college majors. Betz and Hackett found that an individual's Programming self-efficacy plays a major role in deciding students with higher levels of Programming self-efficacy were significantly more likely to choose science-based careers than students with lower levels of Programming self-efficacy.

Programming self-efficacy has also been associated with college students' Programming achievement. In a study of college freshmen, Hall and Ponton (2002) set out to explore the differences between students enrolled in a developmental Programming course and those enrolled in a calculus course. Not surprisingly, the developmental Programming students had lower Programming self-efficacy than the calculus students did. Hall and Ponton hypothesized that this finding supported Bandura's beliefs that Programming achievement is the greatest source of self-efficacy. Developmental Programming students are less likely to have previous successful Programming achievement than calculus students and are therefore less likely to have higher levels of Programming self-efficacy.

Although it has been established that students with higher levels of Programming self-efficacy tend to perform better in Programming, the correspondence between Programming self-efficacy and Programming performance is still not completely understood.

Researchers have also been interested in gender differences in Programming self-efficacy. Unfortunately, research findings have been inconclusive regarding these differences. Some researchers have found a significant difference between the Programming self-efficacy of male and female students, with males demonstrating significantly higher levels of Programming self-efficacy than female students (Betz & Hackett, 1983; Pajares & Miller, 1994). These researchers hypothesized that females' lower levels of Programming self-efficacy were a result of commonly held beliefs that Programming is a male-dominated field or that women are not typically good at Programming. These beliefs lead women to think that they should not be good at Programming, regardless of their actual abilities. In contrast, some researchers have not found gender differences in Programming self-efficacy (Cooper & Robinson, 1991; Hall & Ponton, 2002). In a study of first year students of Hapsburg university, Lent, Lopez, and Bieschke (1991) found a slight difference between the Programming self-efficacy of men and women. They hypothesized that gender differences diminish when male and female students have comparable prior work experiences in Programming. Hackett and Betz (1989) also suggested that the small gender differences they found were due, in part, to gender differences in Programming performances. Because Programming self-efficacy influences students' Programming achievement, researchers have been interested in how accurately students' Programming self-efficacy corresponds to their actual abilities. Hackett and Betz (1989) found that students consistently estimate their abilities inaccurately. The majority of male students' and a large minority of female students' beliefs about their abilities in Programming were incongruent with their actual performances. Similarly, Pajares and Miller (1994) found that students tend to incorrectly estimate their abilities in Programming, with 57% of the students overestimating their abilities and 20% underestimating. Although some overestimates can be

beneficial to help students persist in Programming, underestimates and gross overestimates can be harmful to students' Programming achievement. Researchers have established different sources of Programming self-efficacy, but more research is needed on how students develop inaccurate or misleading self-efficacy in Programming.

### **Debugging anxiety**

Debugging anxiety can also affect students' motivation to learn in Programming classes. Debugging anxiety is related to students feeling tense or anxious when working with code or solving Programming problems (Richardson & Suinn, 1972). Students who suffer from Debugging anxiety do not necessarily experience Debugging anxiety in other subjects. There are many negative consequences of Debugging anxiety. For example, students who experience higher levels of Debugging anxiety typically develop negative attitudes and emotions toward Programming. By the time they reach college Programming courses, their attitudes toward Programming are relatively stable, and students with Debugging anxiety are less likely to take Programming classes or pursue careers requiring Programming. Perhaps the most severe consequence of Debugging anxiety is a decreased level of achievement. Programming performance, Cates and Rhymer (2003) found that students with higher levels of Debugging anxiety had significantly lower computational fluency in all areas of Programming computations. This lower level of fluency in turn decreases students' achievements in Programming and likely contributes to negative attitudes toward Programming. With the clear significance of Debugging anxiety for students' Programming achievement, it is important to consider how researchers go about studying such a complex construct.

### **Measuring Debugging anxiety**

The most widely cited scale used to measure and explore Debugging anxiety is the Debugging anxiety Rating Scale (PDARS) (Suinn, 1972). Researchers suspected that some individuals who did not normally suffer from general Debugging anxiety were still affected by Debugging anxiety, so Suinn developed the PDARS to look more specifically at Debugging anxiety. The purpose of the PDARS was to help researchers explore Debugging anxiety and to evaluate Programming-Debugging anxiety relief techniques. The scale consists of 98 items that address students' Debugging anxiety with the manipulation of numbers and Programming concepts. The PDARS has been used repeatedly by researchers to learn more about how Debugging anxiety affects students and to determine the effectiveness of intervention programs designed to alleviate Debugging anxiety (Bessant, 1995; Capraro, Capraro, & Henson, 2001; Llabre, 1984; Rounds & Hendel, 1980; Zettle & Houghton, 1998).

Multiple researchers have also explored the various dimensions of Debugging anxiety with the PDARS using factor analysis. Rounds and Hendel (1980) found two primary factors of Debugging anxiety when using the PDARS instrument: Programming Test Debugging anxiety and Numerical Debugging anxiety. The items from the PDARS that load onto the Programming Test Debugging anxiety factor deal with the Debugging anxiety students feel before, during, and after Programming tests. Items that load onto the Numerical Debugging anxiety factor are items that cover number manipulation involved primarily in arithmetic. Other researchers have also identified these two main factors through subsequent factor analyses with the PDARS (Alexander & Martray, 1989).

The main criticism of the PDARS is the large number of items on the scale. Several attempts have been made to reduce the number of items by omitting redundant or seemingly irrelevant items (Alexander & Martray, 1989; Levitt & Hutton, 1984; Rounds & Hendel, 1980). Many of the attempts to shorten the PDARS have lacked appropriate generalizability or have failed to

discuss how the items were altered. To establish a valid, shorter version of the PDARS that could be used in a variety of settings, Suinn and Winston (2003) developed a 30-item scale based on the PDARS. The 30 items included on the revised version of the PDARS were selected from the original PDARS scale based on factor loadings in previous studies; items with significant loadings on either the Programming Test Debugging anxiety factor or the Numerical Debugging anxiety factor were included on the revised version. The results of administering the revised version to introductory psychology students established that the revised version is valid, internally consistent, and comparable to the original scale.

One major difficulty in measuring Debugging anxiety is the influence society has on students' beliefs with respect to Programming. In a study on the relationship between Debugging anxiety and social desirability, Zettle and Houghton (1998) found that student's men are less likely than women to report feelings of Debugging anxiety toward Programming because the men believe that it is socially unacceptable for men to experience Debugging anxiety. This finding suggests that researchers will have a more difficult time measuring the Debugging anxiety of male students than that of female students. Zettle and Houghton caution researchers and teachers when using the PDARS in situations that screen students for special intervention programs or other academic opportunities because research suggests that some students might be overlooked because of their unwillingness to truthfully respond to Debugging anxiety questions

Considering the impact Debugging anxiety has on Programming achievement, it is important to ask what causes Debugging anxiety. Jackson and Leffingwell (1999) asked 157 students to reflect on their Programming experiences from elementary school through university. Students were asked to identify challenging experiences in order to explore when they perceived themselves as being stressed in a Programming class. About 27% of the students reported that their first stressful experiences in Programming were at the secondary level. Students consistently identified experiences with their teachers as influencing their emotions and attitudes toward Programming. These experiences included teachers' derogatory comments, negative attitudes and behavior toward their students, and lack of caring about students' understanding. Other researchers have also shown that teachers' attitudes toward both students and the courses they are teaching can influence how students respond to the material (Wilson & Thorton, 2005).

Another factor that contributes to students' Debugging anxiety is the type of instructional method used in the secondary classroom. Clute (1984) explored how two instructional methods, discovery and expository, interacted with students' Debugging anxiety in a high school core curriculum Programming. Clute found that students with higher levels of Debugging anxiety scored higher on the achievement test if they were in the expository format subjects as opposed to the discovery format subjects. On the other hand, students with lower levels of Debugging anxiety performed better in the discovery subject. Clute concluded that there is an interaction between Debugging anxiety and confidence. Students with higher levels of Debugging anxiety would have lower levels of confidence in Programming and therefore would be less likely to perform well in subjects where they would need the confidence to discover the Programming for themselves. Therefore, the amount of confidence required on the student's part to learn the Programming concepts was partially determined by the instructional method.

Little research has been conducted on how student's characteristics, such as gender or elementary school experience, contribute to the development of high school students' Debugging anxiety. Instead, researchers have focused their efforts on how Debugging anxiety

interacts with students' performance. Initially, researchers believed that poor performance led to students feeling anxious about Programming. They hypothesized that students with low achievement in Programming would develop negative emotions and attitudes toward Programming, causing them to avoid Programming in the future (Hembree, 1990). This avoidance would cause students to continue performing poorly, confirming the students' emotions and attitudes toward Programming. Once trapped in this vicious cycle, it would be difficult for students to alleviate their Debugging anxiety without some type of intervention. Although this avoidance cycle is apparent with students who have higher levels of Debugging anxiety, there is little evidence to support low performance initiating or contributing to Debugging anxiety (Ashcraft, 2002). Hoping to find cognitive influences on Debugging anxiety that affect them as they work on Programming problems, Ashcraft and Kirk (2001) investigated how students' working memory affected their levels of Debugging anxiety during Programming tasks. Working memory, or short term memory, is the active part of the memory where information is temporarily stored and manipulated. Ashcraft and Kirk found that Debugging anxiety causes interference with the working memory's ability to focus on the Programming task. For example, students who feel nervous while performing an Programming task will have thoughts about their Debugging anxiety while working on the task. These thoughts take up part of the working memory's capacity, thereby decreasing the amount of working memory available for the Programming task. Ashcraft and Kirk also found that this working memory interference not only causes students to take longer with Programming tasks but also degrades students' accuracy. Closely related to working memory interference is the observation that students often report feeling more anxious about Programming during timed tests (Jackson & Leffingwell, 1999). Walen and Williams (2002) conducted a qualitative study on how two students' Programming performance suffered when time constraints were placed on Programming tests. Although both students demonstrated their understanding of the material in class and on homework, both failed exams in class when they only had a limited amount of time to work. After discussing their concerns about timed tests with their instructors, both were allowed to take the tests without time constraints. Without having to worry about how much time was left both students received excellent grades on their exams and managed to finish within the original time limit. Walen and Williams hypothesized that students' concerns about time interfere with their ability to focus on the Programming tasks at hand. Worrying about the amount of time left can reduce the amount of working memory available to work on the Programming problems.

More research is needed on how Debugging anxiety develops in students and how this development process might differ based on student characteristics, such as gender, ethnicity, classroom experiences, and socioeconomic status. It is difficult for researchers to determine when Debugging anxiety develops because students must be able to recognize when they are feeling anxious and identify the source of their anxieties. Also, as the difficulty of the Programming increases, researchers have a hard time distinguishing between when students are suffering from Debugging anxiety and when they are simply less competent in Programming (Ashcraft & Kirk, 2001). These difficulties need to be taken into consideration when designing studies on Debugging anxiety.

#### **4. RESEARCH METHODOLOGY**

The development process for the questionnaire involved item construction, reliability analysis, and establishing the construct validity of Programming self-efficacy and debugging anxiety items through exploratory factor analysis. Teacher's comments and students interviews are to be used to improve items and interpret the results of the factor analysis.

**Study Area**

George Benson University located in Kalomo district which is about 349km from the capital city of Zambia and about 65km from the capital town of southern province. This school will be sampled due to the many Programming self-efficacy and Debugging anxiety cases recorded and also because it is the nearest to the researcher's residence which will help reduce transport costs.

**Research Design**

The researcher will be descriptive in the sense of collecting information by administering questionnaires and conducting interviews. The investigation will be linked to the Programming self-efficacy and Debugging anxiety questionnaire and also levels of. A survey will be conducted on one school. Questionnaires will be used to collect primary data for 20 percent of the school total population. Respondents will include teachers and students.

**Target Population**

The researcher's survey will be students, teachers and administrators; the current statistics of George Benson university is about 1120 students and there about 125 first year students

**Sample Size and Sampling Procedure**

The researcher will get 20% of the total population of the first year students by using probability sampling.

**Validation and Reliability of Instruments**

A copy of a questionnaire will be handed over to the supervisor for approval and validity. If need be to, possible adjustment will be made based on the supervisor's advice.

**Data Collection Procedures**

A letter will be sent to seek permission to carry out the research at the intended school then the questionnaires will be issued to participants and interviews will be carried out. Because of time, questionnaire will be issued out to participants and later be submitted to the administrator of the school

**Procedures for Data Analysis**

Data collection in this study will be analyzed based on the research findings from different participants and descriptive method will be used to analyze data by using SPSS and EXCEL. Data will be presented using tables, pie charts and percentages. The researcher will analyze data in three stages, firstly, the researcher will transcribe data from structured interviews and questionnaires, secondary data will be balled to identify emerged theme and that will be done using SPSS and EXCEL then presented in tables, pies charts and diagrams. Lastly narration will be done to analyze the results.

**5. CONCLUSIONS**

The questionnaire developed in this study is a reliable and reasonably valid tool for examining first-year students' programming self-efficacy and debugging anxiety. However, it requires further refinement to improve its overall validity, especially construct validity, which is an ongoing process.

The findings show that programming self-efficacy and debugging anxiety are complex and multi-dimensional constructs. Rather than viewing them separately, students tend to perceive both along similar dimensions, such as confidence in assignments, exams, and understanding

of content. This suggests that questionnaires should include diverse items covering multiple aspects of both constructs.

The study also highlights that students' previous programming experiences influence their self-efficacy and anxiety levels. For instance, successful past experiences can increase confidence, while lack of success may reduce it. Therefore, researchers should consider students' backgrounds when designing such tools.

Overall, the study emphasizes the importance of including a wide range of factors in questionnaires to better capture students' beliefs, confidence levels, and anxiety in programming, ensuring a more comprehensive understanding of their learning experiences.

## 6. REFERENCE

- [1] Johnson, & M. P. Zanna (Eds.), *Handbook of attitudes and attitude changes: Basic principles* (pp. 173–222). Mahwah, NJ: Lawrence Erlbaum.
- [2] Alexander, L., & Martray, C. (1989). The development of an abbreviated version of the Debugging anxiety Rating Scale. *Measurement and Evaluation in Counseling and Development, 22*, 143–150.
- [3] Ashcraft, M. H. (2002). Math Debugging anxiety: Personal, educational, and cognitive consequences. *Current Directions in Psychological Science, 11*, 181–185.
- [4] Ashcraft, M. H., & Kirk, E. P. (2001). The relationships among working memory, math Debugging anxiety, and performance. *Journal of Experimental Psychology: General, 130*, 224–237.
- [5] Ashcraft, M. H., & Ridley, K. S. (2005). Math Debugging anxiety and its cognitive consequences. In J. I. D. Campbell (Ed.), *Handbook of mathematical cognition* (pp. 315–330). New York: Psychology Press.
- [6] Bandura, A. (1997). *Self-efficacy: The exercise of control*. New York: W.H. Freeman.
- [7] Beilock, S. L., & Carr, T. H. (2005). When high-powered people fail: Working memory and "choking under pressure" in math. *Psychological Science, 16*, 101–105.
- [8] Beloff, H. (1992). Mother, father and me: Our IQ. *The Psychologist, 5*, 309–311.
- [9] Benson, J., & Nasser, F. (1998). On the use of factor analysis as a research tool. *Journal of Vocational Education Research, 23*, 13–33.
- [10] Bessant, K. (1995). Factors associated with types of Debugging anxiety in college students. *Journal for Research in Programming Education, 26*, 327–345.
- [12] Betz, N. E., & Hackett, G. (1983). The relationship of Programming self-efficacy expectations to the selection of science-based college majors. *Journal of Vocational Behavior, 23*, 329–345.
- [13] Capraro, M. M., Capraro, R. M., & Henson, R. K. (2001). Measurement error of scores on the Debugging anxiety Rating Scale across studies. *Educational and Psychological Measurement, 61*, 373–386. 52
- [14] Cates, G. L., & Rhymer, K. N. (2003). Examining the relationship between Debugging anxiety and Programming performance: An instructional hierarchy perspective. *Journal of Behavioral Education, 12*, 23–34.
- [15] Clute, P. S. (1984). Debugging anxiety, instructional method, and achievement in a survey course in college Programming. *Journal for Research in Programming Education, 15*, 50–58.
- [16] Cooper, S., & Robinson, D. (1991). The relationship of Programming self-efficacy beliefs to Debugging anxiety and performance. *Measurement and Evaluation in Counseling and Development, 24*, 4–11.

- [17] Dobie, T., McFarland, K., & Long, N. (1986). Raw score and factor score multiple regression: An evaluative comparison. *Educational and Psychological Measurement*, 46, 337–347.
- [18] Dowling, D. M. (1978). *The development of an Programming confidence scale and its application in the study of confidence in women college students*. Unpublished doctoral dissertation, Ohio State University, Columbus, OH.
- [19] Eklof, H. (2006). Development and validation of scores from an instrument measuring student test-taking motivation. *Educational & Psychological Measurement*, 66, 643–656.
- [20] Elliott, J. C. (1990). Affect and Programming achievement of nontraditional college students. *Journal for Research in Programming Education*, 21, 160–165.
- [21] Fennema, E., & Peterson, P. L. (1983, April). *Autonomous learning behavior: A possible explanation*. Paper presented at the annual meeting of the American Educational Research Association, Montreal.
- [22] Glynn, S. M., Taasobshirazi, G., & Brickman, P. (2007). Nonscience majors learning science: A theoretical model of motivation. *Journal of Research in Science Teaching*, 44, 1088–1107.
- [23] Gorsuch, R. L. (1983). *Factor analysis*. Mahwah, NJ: Lawrence Erlbaum. Hackett, G., & Betz, N. E. (1989). An exploration of the Programming self-efficacy/Programming performance correspondence. *Journal for Research in Programming Education*, 20, 261–273.
- [24] Hall, J. M., & Ponton, M. K. (2002, March). *A comparative analysis of Programming self-efficacy of developmental and non-developmental freshman Programming students*. Paper presented at the meeting of the Louisiana/Mississippi Section of the Programming Association of America, Natchitoches, LA.
- [25] Hampton, N. Z., & Mason, E. (2003). Learning disabilities, gender, sources of efficacy, self-efficacy beliefs, and academic achievement in high school students. *Journal of School Psychology*, 41, 101–112.
- [26] Hannula, M. (2006). Motivation in Programming: Goals reflected in emotions. *Educational Studies in Programming*, 63, 165–178.
- [27] Hembree, R. (1990). The nature, effects, and relief of Debugging anxiety. *Journal for Research in Programming Education*, 21, 33–46.
- [28] Higbee, J. L., & Thomas, P. V. (1999). Affective and cognitive factors related to Programming achievement. *Journal of Developmental Education*, 23, 8–24.
- [29] Jackson, C. D., & Leffingwell, R. J. (1999). The role of instructors in creating math Debugging anxiety in students from kindergarten through college. *Programming Teacher*, 92, 583–586.
- [30] Kazelskis, R., Reeves, C., Kersh, M., Bailey, G., Cole, K., Larmon, M., Hall, L., & Holliday, D. C. (2000). Debugging anxiety and test Debugging anxiety: Separate constructs? *Journal of Experimental Education*, 68, 137–146.
- [31] Kranzler, J., & Pajares, F. (1997). An exploratory factor analysis of the Programming Self-Efficacy Scale-Revised (PSES-R). *Measurement and Evaluation in Counseling and Development*, 29, 215–228.
- [32] Leeson, H. V. (2006). The mode effect: A literature review of human and technological issues in computerized testing. *International Journal of Testing*, 6, 1–24.
- [33] Lent, R. W., Lopez, F. G., & Bieschke, K. J. (1991). Programming self-efficacy: Sources and relation to science-based career choice. *Journal of Counseling Psychology*, 38, 424–430.

- [34] Levitt, E. E., & Hutton, L. H. (1984). A psychometric assessment of the Debugging anxiety Rating Scale. *Applied Psychology*, 33, 233–242.
- [35] Llabre, M. (1984). Debugging anxiety rating scale. In D. J. Keyser & R. C. Sweetland (Eds.), *Test Critiques: Volume I* (pp. 436–442). Kansas City, MO: Test Corporation of America.
- [36] Lopez, F. G., & Lent, R. W. (1992). Sources of math self-efficacy in high school students. *Career Development Quarterly*, 41, 3–12. Lovell, R. B. (1990). *Adult learning*. New York: Routledge. May, D., & Glynn, S. (2008, February). *An Programming Self-Efficacy Questionnaire for college students*. Paper presented at the annual meeting of Research in Undergraduate
- [37] Programming Education, San Diego. Middleton, J. A., & Spanias, P. A. (1999). Motivation for achievement in Programming: Findings, generalizations, and criticisms of the research. *Journal for Research in Programming Education*, 30, 65–88.
- [38] Morris, L. V., Wu, S. S., & Finnegan, C. L. (2005). Predicting retention in online general education courses. *The American Journal of Distance Education*, 19(1), 23–26.
- [39] Pajares, F., & Graham, L. (1999). Self-efficacy, motivation constructs, and Programming performance of entering middle school students. *Contemporary Educational Psychology*, 24, 124–139.
- [40] Pajares, F., & Kranzler, J. (1995). Self-efficacy beliefs and general mental ability in mathematical problem-solving. *Contemporary Educational Psychology*, 20, 426–443.
- [41] Pajares, F., & Miller, M. D. (1994). Role of self-efficacy and self-concept beliefs in mathematical problem solving: A path analysis. *Journal of Educational Psychology*, 86, 193–203.
- [42] Pajares, F., & Miller, M. D. (1995). Programming self-efficacy and Programming performances: The need for specificity of assessment. *Journal of Counseling Psychology*, 42, 190–198. Pape, S. J., & Smith, C. (2002). Self-regulating Programming skills. *Theory into Practice*, 41, 93–101.
- [43] Parker, A. (1999). A study of variables that predict dropout from distance education. *International Journal of Educational Technology*, 1(2), 56–67.
- [44] Parker, A. (2003). Identifying predictors of academic persistence in distance education. *USDLA Journal*, 7(1). Retrieved on July 15, 2008 from [http://www.usdla.org/html/journal/JAN03\\_Issue/article06.html](http://www.usdla.org/html/journal/JAN03_Issue/article06.html)
- [45] Patton, M. Q. (2002). *Qualitative research and evaluation methods* (3rd ed.). Thousand Oaks, CA: Sage. Pett, M. A., Lackey, N. R., & Sullivan, J. J. (2003). Making sense of factor analysis: The use of factor analysis for instrument development in health care research. London: Sage.
- [46] Pintrich, P. R. (2004). A conceptual framework for assessing motivation and self-regulated learning in college students. *Educational Psychology Review*, 16, 385–407.
- [47] Pintrich, P. R., & Schunk, D. H. (2002). *Motivation in education*. Upper Saddle River, NJ: Merrill. Rammstedt, B., & Rammsayer, T. H. (2000). Sex differences in self-estimates of different aspects of intelligence. *Personality and Individual Differences*, 29, 869–880
- [48] Wigfield, A., Tonks, S., & Eccles, J. S. (2004). Expectancy-value theory in cross-cultural perspective. In D. McInerney & S. van Etten (Eds.), *Research on sociocultural influences on motivation and learning* (pp. 165–198). Greenwich, CT: Information Age.
- [49] Wigfield, A., & Eccles, J. S. (2000). Expectancy-value theory of motivation. *Contemporary Educational Psychology*, 25, 68–81.
- [50] Richardson, R. C., & Suinn, R. M. (1972). The Debugging anxiety Rating Scale: Psychometric data. *Journal of Counseling Psychology*, 19, 551–554.

- [51] Rounds, J. B., & Hendel, D. D. (1980). Measurement and dimensionality of Debugging anxiety. *Journal of Counseling Psychology*, 27, 138–149.
- [52] Schmidt, P. (2007). High school students aim higher without learning more, federal studies find. *Chronicle of Higher Education*, 53, A32.
- [53] Schunk, D. (2004). *Learning theories: An educational perspective* (4th ed.). Upper Saddle River, NJ: Prentice Hall. Shen, C. (2001). Social values associated with cross-national differences in Programming and science achievement: A cross-national analysis. *Assessment in Education*, 8, 193–223.
- [54] Suinn, R. M. (1972). *Debugging anxiety rating scale (PDARS)*. Fort Collins, CO: Rocky Mountain Behavioral Science Institute.
- [55] Suinn, R. M., & Winston, E. H. (2003). The Debugging anxiety rating scale, a brief version: Psychometric data. *Psychological Reports*, 92, 167–173.
- [56] Taylor, J. A., & Galligan, L. (2006). Programming for Programming anxious tertiary students: integrating the cognitive and affective domains using interactive multimedia. *Literacy and Numeracy Studies*, 15, 23–42.
- [57] Usher, E. L., & Pajares, F. (2009). Sources of self-efficacy in Programming: A validation study. *Contemporary Educational Psychology*, 34, 89–101.
- [58] Walen, S., & Williams, S. (2002). A matter of time: Emotional responses to timed Programming tests. *Educational Studies in Programming*, 49, 361–378.