Effect of Self-efficacy and Emotional Engagement on Introductory Programming Students

Geetha Kanaparan
Xiamen University Malaysia
Malaysia
geetha.kanaparan@xmu.edu.my

Rowena Cullen
Victoria University of Wellington
New Zealand

David Mason
Victoria University of Wellington
New Zealand

Abstract

High failure rates appear to be a norm in introductory programming courses. Many solutions have been proposed to improve the high failure rates. Surprisingly, these solutions have not led to significant improvements in the performance of students in introductory programming courses. In this study, the relationship between self-efficacy, emotional engagement and the performance of students in introductory programming courses were examined. Enjoyment, interest, and gratification were identified as three factors contributing to emotional engagement in introductory programming courses from a review of existing literature and from focus groups. An online survey of 433 students in introductory programming courses showed that the students' programming self-efficacy beliefs had a strong positive effect on enjoyment, while gratification and interest had a negative effect on programming performance. These findings have implications for course instructors who design and deliver introductory programming courses.

Keywords: self-efficacy; emotional engagement; introductory programming courses; programming performance

1 Introduction

Software developer jobs are in demand due to the rapid advances in technology that has enabled automation in many businesses today. A recent report by the Australian Government showed that as of November 2017, Software and Application Programmers were amongst the top three most in demand occupations within the Professional, Scientific and Technical Services industry which itself was one of top five growing industries in Australia (Australian Government Department of Jobs and Small Business, 2018). In yet another report, the U.S. Department of Labour reported that employment for software developers is expected to increase by 24% from 2016 to 2022 (Bureau of Labour Statistics, 2018).

Higher educational institutions (HEIs) play an important role in attracting, retaining, and producing graduates who are qualified in Information and Communications Technology (ICT) related disciplines in order to meet the rising demand for software developers. However, failure rates of between 30% and 50% (Bennedsen & Caspersen, 2007; Quille & Bergin, 2016;

Watson & Li, 2014) in introductory programming courses poses a problem to HEIs in meeting the demands of a skilled workforce for the software industry.

Introduction to programming is a core course in the first year of an Undergraduate ICT related discipline, introducing students to the fundamental concepts of computer programming and equips them with the skills to code solutions to programming problems. Success in the introductory programming course is crucial so that students are able to understand and write programming code, and it is a mandatory course for students who wish to obtain a qualification in an ICT related discipline and pursue a career as a software developer.

Kanaparan, G et al (2013) argued that high failure rates may be due to the demanding cognitive load of programming languages and the behaviour of the students. Although a large number of solutions have focused on reducing the demanding cognitive load of programming languages, there has been little to no success in reducing the high failure and attrition rates. By contrast, the behaviour of the students has not been examined extensively although the literature on learning theory suggests that there is a strong link between the behaviour of the student and their performance (Bartimote-Aufflick, Bridgeman, Walker, Sharma, & Smith, 2016; Schunk & Mullen 2012).

Despite a dearth of research on the behaviour of students in introductory programming courses, the study by Wiedenbeck, Xiaoning, and Chintakovid (2007) provided motivation for this research. The authors found that a student's self-efficacy in computer programming affects their computer interest, and that their computer interest then affects their programming performance. Computer interest was conceptualised as "end users who have interest in computers, programming, and related topics, even if they have low experience" (Wiedenbeck et al., 2007, p. 70). Interest is one of several indicators of engagement and is discussed in the next section. Despite the encouraging findings, no further research could be found in the literature on computer programming that examines the link between self-efficacy, engagement, and performance.

By contrast, the literature on learning theory suggests that self-efficacy is one contextual factor that affects engagement, and engagement then affects the performance of the student (Appleton, Christenson, & Reschly, 2006; Linnenbrink & Pintrich, 2003; Schunk & Mullen, 2012). Although self-efficacy is necessary for success in learning, the student may not necessarily be engaged in learning (Appleton et al., 2006). Instead, students who engage in learning are self-efficacious learners, strive to achieve their goals, and are ultimately successful (Schunk & Mullen, 2012). Recognising the importance of self-efficacy, several researchers have produced frameworks to explain the relationship between self-efficacy, student engagement and learning, a discussion of which is provided in the next section.

Due to the compelling evidence presented in the literature on learning theory, this research examines two behavioural factors: self-efficacy and emotional engagement, and the effect of these two behavioural factors on the performance of students in introductory programming courses.

This research asks the following questions:

Research Question 1: What is the effect of programming self-efficacy on the emotional engagement of students in introductory programming courses?

Research Question 2: What is the effect of emotional engagement on the programming performance of students in introductory programming courses?

2 Theoretical Framework and Research Hypotheses

2.1 Programming self-efficacy and Social Cognitive Theory

Social Cognitive Theory (SCT) is a widely accepted and empirically validated framework that explains human behaviour. The SCT framework is based on the premise that human beings have the ability to control their thoughts, feelings, motivation, and actions (Bandura, 1986). Bandura refers to this ability as a self-system which allows human behaviour to be understood, predicted, and altered within a social context (Bandura, 1977a; Bandura, 1986). SCT explains human behaviour based on three interacting determinants which form a triadic reciprocality (Figure 1). The three interacting determinants are: personal factors, behaviour, and the environment. The personal factors determinant is shaped by expectations, beliefs, goals, self-perceptions and intentions; while the environmental determinant refers to factors such as social norms, access in the community and influence on others, and the behavioural determinant refers to skills such as practise and self-efficacy (Bandura, 1986; Pajares, 1997).

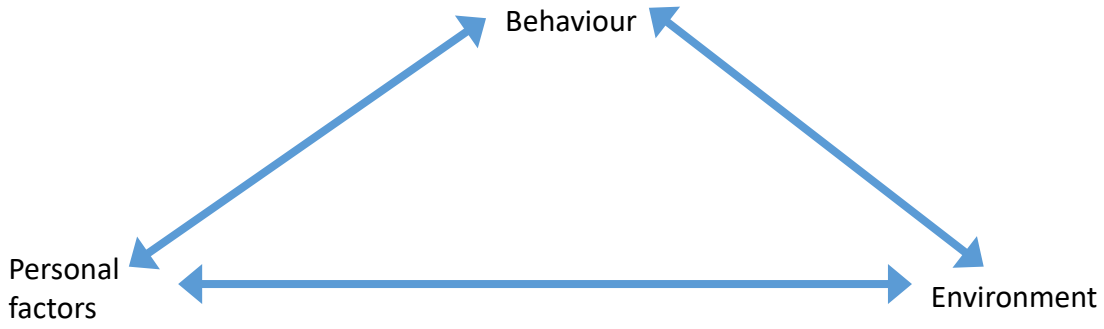


Figure 1: Three-way relationship in triadic reciprocality (Bandura, 1986, p. 24)

As an example, a programming student who encounters an error while working on a programming project might seek help from peers or a teacher (personal factors influences behaviour). The teacher or peer proceeds to explain or debug the error with the student (behaviour influences environment). The student then reviews the error and understands how the error was resolved (environment influences personal factors). This suggests that the interaction between personal factors, behaviour, and the environment influences an individual's self-belief, which then affects the environment and subsequently affects the individual's performance (Pajares, 1997).

The theory of self-efficacy forms part of Bandura's SCT and was proposed to predict and explain behavioural change in individuals. Self-efficacy is defined as "people's judgments of their capabilities to organize and execute courses of action required to attain designated types of performances" (Bandura, 1986, p. 391). The definition of self-efficacy suggests that the individual's perception of their ability may influence how well they carry out a task or activity, and that their ability then determines the degree of success in completing the task or activity. High self-efficacy in a task implies that the individual believes that they have the ability to accomplish that task successfully (Walker, Greene, & Mansell, 2006) and that self-efficacy influences an individual's choice and action (Schunk, 1996). Further, an individual with high

self-efficacy beliefs is likely to engage in tasks or activities that they feel they are competent in and are confident in accomplishing. Conversely, an individual with low self-efficacy beliefs avoids tasks that they feel they are not able to do, perceiving them as difficult to accomplish (Schunk, 1996). Students with high self-efficacy demonstrate a positive attitude towards learning by seeking challenges, become intrinsically interested, have set goals, persist in the face of challenge, adopt effective strategies to resolve challenges, and are able to easily recover from failures or setbacks (Bandura, 1997; Schunk, 1996).

Existing research on learning theory largely examined the effect of self-efficacy on student motivation, learning and performance, and found that self-efficacy beliefs affects performance, perseverance to learn, whilst also having the ability to alter behaviour (Phan, 2011; Yip, 2012; Zimmerman, 2000). By contrast, research on the effect of self-efficacy on programming students appears to be limited. One such research identified the factors that influence the self-efficacy beliefs of students in introductory programming courses (Zingaro, 2014) while two other researchers reported positive outcomes when examining the relationship between the student's self-efficacy beliefs and their programming performance (Kinnunen & Simon, 2011; Ramalingam, LaBelle, & Wiedenbeck, 2004).

Self-efficacy is proposed as an independent variable in this research due to the strong evidence presented in the literature on learning theory about the positive influences of self-efficacy on learning and performance. This study proposes to use the term programming self-efficacy in reference specifically to the student's judgment of their ability to learn programming.

2.2 Emotional Engagement and Student Engagement

Student engagement has emerged as a theoretical model for understanding which students are likely to fail, dropout from school, and for improving student motivation and achievement (Appleton, Christenson, & Furlong, 2008; Christenson, Reschly, & Wylie, 2012; Fredricks, Blumenfeld, & Paris, 2004). Engagement is a multi-dimensional construct and is defined as "the student's active participation in academic and co-curricular or school-related activities, and commitment to educational goals and learning.... student engagement drives learning; requires energy and effort; is affected by multiple contextual influences; and can be achieved for all learners" (Christenson et al., 2012, pp. 816-817).

Over the years, several student engagement models and frameworks have been proposed. There is an on-going debate for a commonly accepted model for student engagement (Christenson et al., 2012) due to the large body of research on the dimensions of student engagement (Fredricks et al., 2004) and the lack of agreement between researchers on the three schools of thought that make up the literature on student engagement (Reschly & Christenson, 2012). One school of thought arose as a result of dropout prevention theory and intervention, the second from general school reform, and the third school of thought arose from motivational literature. Within these schools of thought, sub-disciplines were formed with interests in various aspects of engagement. For example, educational psychologists are interested in examining engagement within an educational context while development psychologists are interested in engagement from the perspective of motivational theory (Reschly & Christenson, 2012).

Despite the lack of a commonly accepted engagement model, the more frequently discussed student engagement models include Linnenbrink and Pintrich's framework (Linnenbrink & Pintrich, 2003), Appleton et al.'s Student Engagement Model (Appleton et al., 2006), and

Reschly and Christenson's Student Engagement Model (Reschly & Christenson, 2012). These models and framework vary in the dimensions of engagement, indicators used to measure student engagement, the range of contextual factors that affect student engagement, and the outcomes of student engagement.

However, there is general agreement among researchers that there are three dimensions of engagement. Two of the engagement dimensions are behavioural engagement and cognitive engagement (Appleton et al., 2006; Fredricks et al., 2004; Linnenbrink & Pintrich, 2003; Reschly & Christenson, 2012), and the third dimension proposed was motivational engagement (Linnenbrink & Pintrich, 2003), emotional engagement (Fredricks et al., 2004), psychological engagement (Appleton et al., 2006), or affective engagement (Betts, Appleton, Reschly, Christenson, & Huebner, 2010). Despite a lack of consensus on the third engagement dimension, emotional engagement appears to have been more commonly discussed, examined, and clearly conceptualised in the literature on learning theory.

Emotional engagement is a dependent variable in this research since Wiedenbeck et al. (2007) argued that computer interest (an indicator of emotional engagement) is an important engagement factor when learning programming. Emotional engagement refers to the student's feeling, attitude, perception towards learning, and the learning environment (Sheard, Carbone, & Hurst, 2010; Yazzie-Mintz & McCormick, 2012). In this study, enjoyment and interest were identified as indicators of emotional engagement after an extensive review of the literature on learning theory (Appleton et al., 2006; Linnenbrink & Pintrich, 2003; Reschly & Christenson, 2012) and from the literature on computer programming. Enjoyment and interest were then validated as proposed indicators of emotional engagement in introductory programming courses through focus groups, while gratification emerged as a third indicator of emotional engagement during the focus groups.

Other factors that were identified and considered as potential indicators of emotional engagement include: affect, value, belonging, and identification with school membership/connectedness. Affect did not appear to be a suitable indicator because the conceptualisation of affect had appeared to collectively measure several indicators of emotional engagement that had been identified in this study. Similarly, value, belonging, and identification with school membership/connectedness did not appear to be suitable indicators in this study because these indicators were better measures of the student's perception of the educational institution that they were attending instead of their performance in their introductory programming course.

2.2.1 Focus Groups – Validating Indicators of Emotional Engagement

To validate the proposed indicators of emotional engagement in introductory programming courses, four (4) focus groups were conducted in New Zealand and in Malaysia. In total, 16 students who were mid-way through their introductory programming course participated in the focus groups. There were between 3 and 5 participants in each focus group. The duration for each focus group was between 45 and 60 minutes. The discussion in the third and fourth focus groups did not reveal any new findings that were not already identified in the first two focus groups. As such, it was felt that no further focus groups were required.

The questions were reviewed by three academics who had more than 15 years of teaching experience at the tertiary level. The students were asked questions that encouraged them to

discuss how they felt when they were learning programming and when they were working on their programming assessments.

The focus groups were transcribed and analysed. The analysis of the focus groups conforms to the key concepts analysis framework that identifies the key concepts or themes in the research by asking the participants to identify the core themes, as well as the important ideas and experiences that relate to these themes (Krueger & Casey, 2009). The data from the focus groups were analysed for evidence of the proposed emotional engagement indicators by examining the engagement behaviour of the students. Next, an inductive and deductive approach was adopted to analyse the data (Thomas, 2006). The deductive approach was used to examine the data for evidence of the indicators of emotional engagement, while the inductive approach was used to code the data, and to identify new emotional engagement indicators. The findings from the analysis of the focus groups confirmed that enjoyment and interest were indicators of emotional engagement in introductory programming courses, and interestingly, gratification emerged as a third indicator of emotional engagement in introductory programming courses.

2.2.2 Enjoyment

Enjoyment "causes the subject to experience pleasure by causing occurrent beliefs which satisfy desires concerning the experience itself" (Davis, 1982, p. 240). The following are quotes from the participants in the focus groups who appeared to enjoy learning programming:

> *"Because I really enjoy programming. Maybe I go so far as to say I have fun while doing it." (CL, FG2)*
>
> *"…thoroughly enjoy programming... thoroughly hate doing essays" (CM, FG1)*
>
> *"I actually enjoy doing this subject [referring to programming course] and I want to like start on it [new programming assessment]" (JR, FG3)*

In the literature on learning theory, positive correlations were observed between self-efficacy and enjoyment (Mills, Pajares, & Herron, 2007), and that high self-efficacy results in pleasant emotions such as enjoyment (Putwain, Sander, & Larkin, 2013). Therefore, the following hypothesis is proposed:

H1: Self-efficacy beliefs in learning programming will have a positive effect on enjoyment.

The literature on computer programming suggests that enjoyment in learning programming increases as a result of using the Alice programming environment (Bishop-Clark, Courte, Evans, & Howard, 2007) and when engaging in pair programming (Liebenberg, Mentz, & Breed, 2012; Maguire, Maguire, Hyland, & Marshall, 2014). However, the effect of enjoyment on the performance of the student was not examined in these studies. By contrast, Frenzel, Thrash, Pekrun, & Goetz (2007) found positive correlations between enjoyment and the academic performance of students. Therefore, the following hypothesis is proposed:

H2: Enjoyment in learning programming will have a positive effect on programming performance.

2.2.3 Interest

Interest is defined as "an emotion that arouses attention to, curiosity about, and concern with…" a discipline of study (Akbulut & Looney, 2007, p. 68). The following are quotes from the participants in the focus groups who appeared to have an interest in learning programming:

"I am interested because the code itself makes me very curious about what the outcome would be" (FA, FG4)

"…programming is going to be my life…that's what I want to be when I'm finished..." (MH, FG1)

"I am one of those (who play games)….. Super Mario (game) doing certain things... start wondering what logic lies behind it... what codes... and who thought about it first? So, you start wondering, let's poke something here and see what comes out." (MJE, FG4)

Silvia (2003) suggested that self-efficacy affects interest indirectly, whereby "self-efficacy affects uncertainty about how the activity will resolve, which in turn affects interest" (p.239). In another study, Linnenbrink and Pintrich (2003) argued that as the student develops expertise in a given task, his or her self-efficacy beliefs and interest increases. Additionally, Bandura (1997) suggests that a moderate level of self-efficacy is essential for sustaining interest in a given task, while Akbulut and Looney (2007), and Wiedenbeck et al. (2007) found a positive relationship between programming self-efficacy and interest. Therefore, the following hypothesis is proposed:

H3: Self-efficacy beliefs in learning programming will have a positive effect on interest.

McKinney and Denton (2004) and Wiedenbeck et al. (2007) found that interest is significantly correlated to programming performance, while Sheard et al. (2010) reported that students have a high interest in topics such as programming and computer networks. Furthermore, in the literature on learning theory, Bye, Pushkar, & Conway (2007) found that interest is a strong predictor of motivation to learn. Therefore, the following hypothesis is proposed:

H4: Interest in learning programming will have a positive effect on programming performance.

2.2.4 Gratification

Gratification emerged as a new indicator of emotional engagement during the focus groups. In the literature on learning theory, research on gratification discusses "delay of gratification" and its effect on academic success. Delay of gratification is the "voluntary postponement of immediate rewards and persistence in goal-directed behaviour for the sake of later outcomes" (Mischel, Schoda, & Rodriguez, 1989, p. 933).

However, during the focus groups, the participants felt a feeling of immediate gratification when they were able to debug the errors in their program and were able to see the output of their program. The following are quotes from the participants in the focus groups who appeared to have a feeling of immediate gratification when learning programming:

"..Just kind of smile thinking you've done it. It's finished." (DI, FG2)

"...especially if I figure it out for myself that's a really rewarding feeling." (RS, FG2)

"…very satisfying feeling when it works… it doesn't, it's very frustrating..." (CM, FG1)

"Not frustration but definitely excitement when I finish [my programming assessment]." (RB, FG3)

"Triumph [when I find a solution to my programming problem]" (CL, FG2)

For example, participant DI from Focus Group 2 explained that programming gave him an immediate sense of achievement compared to other courses. Similarly, participant RS from Focus Group 2 felt rewarded each time she achieved a milestone in her programming

assessment but added that she did not feel the same way after she had completed the assessments for other courses. Participant CL from Focus Group 2 echoed similar feelings of gratification whereby he felt triumphant when he was able to solve a programming problem.

Therefore, for the purpose of this study, gratification is defined as students who experience "the feeling of receiving immediate rewards typically in the form of pleasure or satisfaction as a result of hard work". The interpretation of gratification in this study focuses on *immediate gratification* as opposed to *delay of gratification* which is a common outcome of learning in the literature on learning theory. The following hypotheses are proposed despite a lack of evidence from the existing research that suggests a relationship between programming self-efficacy and gratification, and a relationship between gratification and programming performance:

H5: Self-efficacy beliefs in learning programming will have a positive effect on gratification.

H6: Gratification in learning programming will have a positive effect on programming performance.

2.3 Programming Performance: The Dependent Variable

The dependent variable in this research is programming performance. The term programming performance has been used widely in the literature on computer programming. Although no specific definition could be found in the literature, an analysis of the attribute(s) used to measure programming performance suggests that programming performance refers to an objective measure of how well the student has performed in their introductory programming course. The programming performance of the students were measured by the final grade that the students received at the end of their introductory programming course. Grades are assigned to each piece of assessment that the student submits in the introductory programming course and these grades cumulatively contribute to the final grade of the student's programming performance.

Two concerns arose when deciding on the use of the final grade as a measure for programming performance. Firstly, the final grade may not reflect the true programming ability of the student, and secondly, the assessment design in some introductory programming courses may include non-programming related tasks or activities. In order to mitigate these concerns and to validate the use of programming grade to measure programming performance, the course outline was obtained from the instructors of each introductory programming course. The course outline was then examined for the method of assessment and the learning outcomes of the course. It was found that the assessments ranged from programming assignments, class tests, to lab-based exercises which were clearly linked to the learning outcomes of the introductory programming course. This served to allay the aforementioned concerns thereby giving confidence that the assessments were designed to test the programming ability of the students. As such, there was strong evidence that the final grade was able to provide an objective measure of the student's level of performance in the introductory programming course. Additionally, evidence from existing research on introductory programming courses suggests that course grade is a strong indicator of success and is a criterion for progression to the next level of study (deBry, 2011; Ford & Venema, 2010).

2.4 Confounding Variables

Confounding variables are variables that may be "mixed up with the independent variable, making it impossible to determine which of the variables has produced changes in the dependent variable" (Stangor, 2011, p. 231). Prior programming experience and learning

ability were two confounding variables identified from the literature on computer programming, that were likely to influence the programming performance of students. Learning ability was measured by the high school/pre-University results of the students while prior programming experience was measured by the participant's evaluation of their programming experience. The results showed that the confounding variables did not significantly affect the performance of the students.

2.5 Research Model

Figure 2 shows the research model for self-efficacy, (indicators of) emotional engagement and programming performance, and their hypotheses.

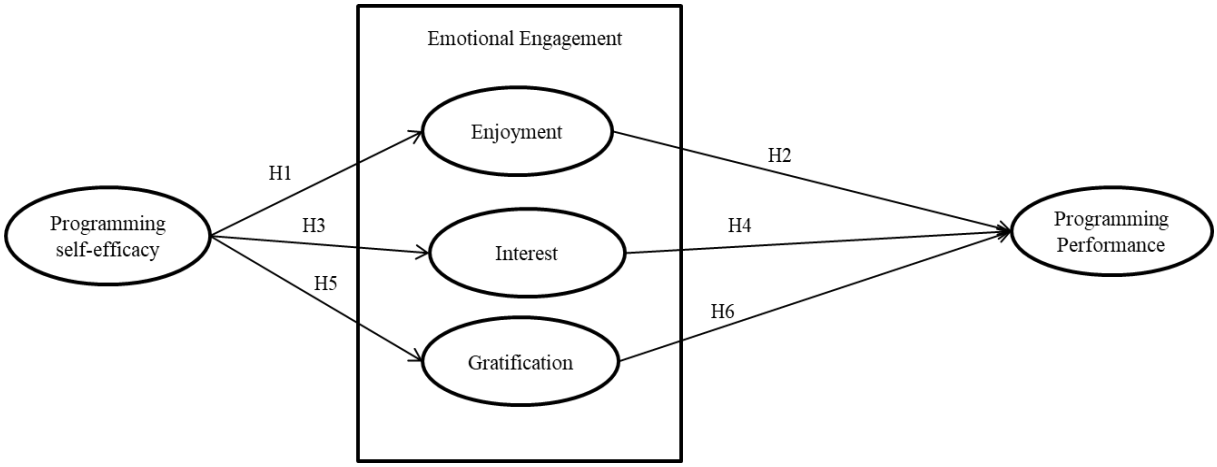


Figure 2. Research model for self-efficacy, emotional engagement, and programming performance

The hypotheses on the left side of the research model (H1, H3 and H5) seeks to address Research Question 1, and the hypotheses on the right side of the research model (H2, H4 and H6) seeks to address Research Question 2.

3 Methodology

The data for programming self-efficacy, emotional engagement, prior programming experience (confounding variable) and the demographic information of the participants was collected using a cross-sectional survey. Self-reporting questionnaires were administered to students in the final week of their introductory programming course. The programming grade and high school/pre-University results (confounding variable) of the survey participants were retrieved from their institution's student records after obtaining approval from the participants. This was necessary to ensure the integrity of the data. The signed Participant Consent Form was sent to the Manager responsible for student records as evidence of consent to access the data.

3.1 Survey instrument

The items to measure programming self-efficacy was adapted from existing empirical research on programming self-efficacy (Ramalingam & Wiedenbeck, 1998), academic self-efficacy (Bresó et al., 2011; Sherer at al., 1982) and from Bandura's (1986) three dimensions of self-efficacy: magnitude, strength, and generality. These sources of item development conform to the recommendations made by MacKenzie et al. (2011) for the development of measurement items. The reliability of the items to measure programming self-efficacy was initially reviewed by five introductory programming course instructors and were re-worded to make them

relevant to the context of programming and for this research. A 5-point Likert scale was used, ranging from 1 (not confident at all) to 5 (very confident).

The items to measure interest and gratification were developed from the findings of the focus groups and from the definition of the construct. At the time of the development of the instrument, no existing literature could be found with access to the items that measure interest and gratification. By contrast, the items to measure enjoyment were adapted from existing research that was conducted by Bishop-Clark et al. (2007) and from the focus groups. Here again, a 5-point Likert scale was used to measure emotional engagement, ranging from 1 (Strongly Disagree) to 5 (Strongly Agree).

The survey instrument was further subjected to rigorous reliability and validity tests. A card sorting activity was performed to assess construct validity and to identify items that lacked clarity (Moore & Benbasat, 1991). A pilot study was then conducted, and finally an Exploratory Factor Analysis (EFA) was performed to assess the convergent and discriminant validity of the items in the survey. The survey items are listed in Appendix A.

3.2 Data collection

The data was collected in two countries, Malaysia and New Zealand, which were able to offer representativeness of the sample and an international perspective to this study. A total of 10 tertiary level educational institutions participated in the survey, 6 of which were from New Zealand, and 4 were from Malaysia. The questionnaire was designed and administered online. Table 1 shows the response rate by country.

Country	Sample Population	No. of Participants	No. of Usable Responses
New Zealand	750 (68.6%)	225 (47.9%)	214 (49.4%)
Malaysia	343 (31.4%)	245 (52.1%)	219 (50.6%)
Total	1093	470	433

Table 1. Response Rate by Country

A total of 1093 students were invited to participate in the survey. The response rate was high: 43% (470) of the students from the sample population participated in the survey, of which 92.1% (433) of the responses were usable. Out of the 433 usable responses, 72.7% were male participants and 33% (144) of the participants had prior programming experience. However, only 1.7% (14) of the participants had scored themselves as having good to extensive programming experience.

The programming grades and high school/pre-University results were obtained between 6 and 10 weeks after the students completed their final programming assessment. Table 2 shows the distribution of the final grades obtained by the participants in their introductory programming course.

Grade	Malaysia	New Zealand	No. of students (%) (n = 433)
A (80% - 100%)	45	117	162 (37.4%)
B (65% - 79%)	76	59	135 (31.2%)
C (50% - 64%)	69	29	98 (22.6%)
D (40% - 49%)	19	5	24 (5.6%)
E (< 40%)	10	4	14 (3.2%)

Table 2. Distribution of Programming Grade by Country

Clearly, more than half of the participants (68.6%) had achieved a grade B or better in their introductory programming course, while only 8.8% of the participants had obtained a grade D or poorer. Interestingly, a significantly higher percentage of the participants in New Zealand had obtained a Grade A (72.2%) compared to Malaysia (27.8%).

4 Results

The data was analysed using the variance-based PLS-SEM SmartPLS 3 software (Ringle, Wende, & Becker, 2015) since the objective of this research is to predict and explain the constructs, and PLS-SEM is able to handle single-item constructs (Hair, Hult, Ringle, & Sarstedt, 2014) such as programming performance and learning ability. The analysis of the data was performed in two stages. The first stage involved the analysis of the measurement model, and the second stage involved the analysis of the structural model (Hair et al., 2014). The measurement model describes the relationships between the constructs and their corresponding indicators (Hair et al., 2014) by establishing the reliability and validity of the constructs. The path connecting the items to the constructs were measured reflectively. The Composite Reliability (CR), indicator reliability (item loadings), and Average Variance Extracted (AVE) were established for the measurement model (Hair et al., 2014). Table 3 presents the convergent validity of the constructs in the research model. A CR value of 0.7, AVE value of 0.5, and item loadings that are greater than 0.6 are acceptable to satisfy the convergent validity of the constructs (Hair et al., 2014).

Construct	No. of Items	Item Loadings	AVE	CR
Programming self-efficacy	22 items	.622 – .770	.511	.958
Enjoyment	3 items	.680 - .866	.625	.832
Interest	3 items	.684 - .890	.675	.860
Gratification	4 items	.618 - .875	.590	.850
Programming Performance	1 item	1.000	-	-

Table 3. Convergent Validity

The discriminant validity of the measurement model was then examined. Discriminant validity refers to how well a construct differs from the other constructs in the model through

statistical comparisons (Hair et al., 2014). To satisfy the conditions for discriminant validity, firstly, the results of the Fornell-Larcker criterion and the cross-loadings of the items were examined. When evaluating the discriminant validity using the Fornell-Larcker criterion, the square root of a construct's AVE should be higher than any of the correlation with other constructs (Fornell & Larcker, 1981). Table 4 presents the results for discriminant validity using the Fornell-Larcker criterion.

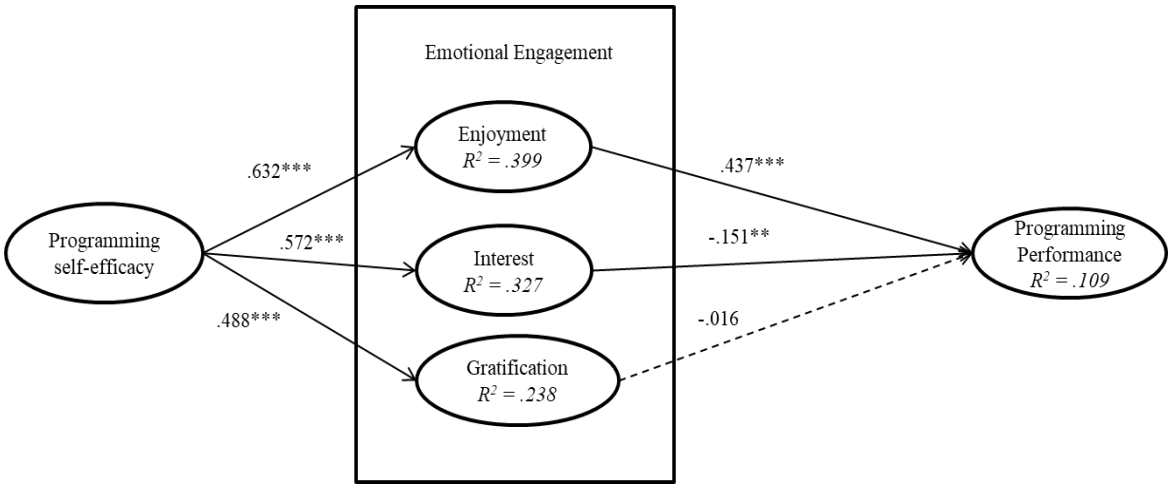
Construct	Programming self-efficacy	Enjoyment	Interest	Gratification	Programming Performance
Programming self-efficacy	**.715**				
Enjoyment	.632	**.790**			
Interest	.572	.753	**.821**		
Gratification	.488	.542	.519	**.768**	
Programming Performance	.413	.314	.170	.142	**1.000**

Table 4. Discriminant Validity

The diagonal values that are highlighted in bold text represent the square root of the AVE, while the off-diagonals represent the correlations. The results for the Fornell-Larcker criterion confirm that the square root of a construct's AVE is higher than the correlations of the other constructs. The second condition for discriminant validity requires that the items should load more strongly on their construct than on other constructs (Hair et al., 2014). The cross-loadings of the items were examined and found to load strongly on their intended construct than on other constructs. Thus, the conditions for discriminant validity have been met.

The structural model tests the hypotheses in the research model and describes the relationships between the constructs (Hair et al., 2014). Figure 3 presents the structural model with the results of the path analysis. A bootstrapping procedure was used to determine the significance of the path coefficients (Chin, 1998). Five thousand (5000) bootstrap samples were used to estimate the PLS path model (Hair et al., 2014). The R2 values for the constructs determine the predictive accuracy of the model. The R2 values for enjoyment and interest constructs exceeded the moderate threshold predictive accuracy that was proposed by Chin (1998), while the R2 values of the gratification construct and the programming performance construct were weak.

The results of the path analysis in Figure 3 shows that relationships H1 (programming self-efficacy -> enjoyment), H3 (programming self-efficacy -> interest), and H5 (programming self-efficacy -> gratification) were positive and significant.



Note: t-values > 1.65 (p<0.1); t-values > 1.96** (p<0.05); t-values > 2.57*** (p<0.01)*
Figure 3. Structural model with results of path analysis

A strong positive and significant relationship was found in path H2 (enjoyment -> programming performance), while the result of the path analysis for H4 (interest -> programming performance) showed a negative but statistically significant relationship. However, hypothesis H6 (gratification -> programming performance) showed a weak negative and not statistically significant relationship. Therefore, Hypothesis H6 was not supported.

The research model was tested for common method bias since the data for the independent and the dependent variables was collected using a single method (Podsakoff, MacKenzie, Lee, & Podsakoff, 2003). An Exploratory Factor Analysis (EFA) was performed on all the measurement items in order to control for common method bias. Common method bias may exist in a dataset if the majority (>50%) of the items load on one factor (Podsakoff & Organ, 1986). The results of the EFA showed that the largest variance explained by an individual factor was 24.04%, suggesting that common method bias was not a problem in this research.

5 Discussion

This study proposed and tested a research model on programming self-efficacy, emotional engagement, and programming performance. Enjoyment, interest, and gratification were identified as three indicators of emotional engagement in introductory programming courses. Several key findings emerged from this research.

Our first finding suggests that the strong link between programming self-efficacy and emotional engagement validates the importance of self-efficacy beliefs on human behaviour, and the importance of Bandura's Social Cognitive Theory within the context of students in introductory programming courses.

Second, the positive relationships between programming self-efficacy and enjoyment, and between programming self-efficacy and interest supports the findings from the published literature on learning theory.

Third, the positive relationship between programming self-efficacy and gratification is strong evidence that gratification is an indicator of emotional engagement and that it appears to be an important engagement factor in the programming discipline although no published research could be found to support this relationship.

Fourth, the results showed a positive relationship between enjoyment and the programming performance of the students. This result supports the findings from the published literature on learning theory that found positive correlations between enjoyment and performance (Frenzel et al., 2007), and that enjoyment is an important emotional engagement factor that leads to better programming performance in introductory programming courses.

Fifth, the negative relationship between interest and programming performance did not support our initial hypothesis and contradicts existing research which found positive correlations between interest and programming performance (McKinney & Denton, 2004; Wiedenbeck et al., 2007). O'Keefe and Linnenbrink-Garcia (2014) argued that students who show interest in performing a task may be doing so due to affect-related interest, while students who place importance in performing a task well may be demonstrating value-related interest. In this study, it may be possible that the research participants had high affect-related interest but low value-related interest. As a result, the research participants may have placed more importance on being interested in learning to program and had not considered the criteria for performing well in their introductory programming course.

Lastly, despite a lack of evidence in the literature, the relationship between gratification and programming performance was hypothesised as a positive relationship. However, the findings in this research revealed a weak negative relationship between gratification and programming performance. This finding suggests that although students felt gratified upon seeing their program work without errors, their actual performance may not have been satisfactory. One plausible explanation could be that although the students were able to see the output of their program and were able to make their program work without errors, they may not have met all the requirements of the programming assessment that would lead to a better programming grade.

These findings have implications for course instructors who are involved in the design and delivery of introductory programming courses. We argue that the cognitive demands of learning programming may be managed by building self-efficacy beliefs, by embedding enjoyable learning activities, and by encouraging affect-related interest while simultaneously developing the value-related interest of the students. Table 5 summarises the implications and recommendations for course instructors.

Implication	Recommendation
Build self-efficacy beliefs to manage cognitive demands	• Weekly programming exercises to encourage practice • Provide feedback on exercises • Design programming tasks around threshold concepts
Embed enjoyable learning activities	• Encourage pair programming • Use new programming tools • Apply gamification
Encourage affect-related interest, develop value-related interest	• Provide clear guidelines and explain criteria for performing well

Table 5. Implications and recommendations for course instructors

Social Cognitive theorists argue that self-efficacy beliefs may be developed through enactive mastery, vicarious experiences, verbal persuasion, and/or physiological cues (Bandura, 1997). One way to apply the enactive mastery approach is by introducing weekly programming

exercises that would enable students to practice writing programs so that they are able to improve their understanding. The feedback provided by the course instructor on the weekly programming exercises would enable the student to understand the source of the errors and how to overcome them, leading to increased self-efficacy beliefs to learn programming.

In particular, course instructors should focus on designing programming tasks around threshold concepts, which is one factor that contributes to the difficulties of learning programming. Threshold concepts are troublesome concepts that may hinder learning as the learner may become stuck, frustrated, lose interest in the subject, adopt a surface approach to learning, or even withdraw from the course (Boustedt et al., 2007; Sorva, 2010) which could result in low self-efficacy beliefs. Threshold concepts are typically difficult to master, but once the programming concept is understood, students will be able to progress to the next programming concept. By focusing on programming tasks that require students to repeatedly master these threshold concepts, students may be able to increase their self-efficacy beliefs in programming. Furthermore, students will be able to see immediate results upon completing their programming task and thus experience a feeling of gratification. The feeling of gratification is an indicator of emotional engagement unique to this study, and not found in other studies related to learning programming.

Evidence from the focus groups suggests that the challenge of learning programming creates a sense of enjoyment in students. In order to support students who may feel discouraged by the challenges of learning programming, course instructors could embed learning activities that are intended to invoke the feeling of enjoyment. Using pair programming (Liebenberg et al., 2012; Maguire et al., 2014), introducing new programming tools (Bishop-Clark et al., 2007) and applying gamification techniques (Fotaris, Mastoras, Leinfellner, & Rosunally, 2016) are some examples from the extant research on computer programming that have successfully engaged students in enjoyable activities when learning programming.

Interest in programming clearly has an effect on the performance of the students. Students who show affect- related interest in programming should continue to be encouraged and should simultaneously be guided to develop their value-related interest. Course instructors could develop the value-related interest of the students by providing clear guidelines and explaining the criteria for performing well in their programming assessment

6 Conclusion

This research examined the relationship between programming self-efficacy and emotional engagement, and the relationship between emotional engagement and the programming performance of students in introductory programming courses. We argue that the results of this study may be generalised to the larger population of students in introductory programming courses as data was collected from a fairly large (433 students) heterogeneous group of participants. Additionally, data was collected from several institutions that offer tertiary level education in Malaysia and in New Zealand with students of various nationalities enrolled in their introductory programming course.

Gender imbalance and an uneven distribution in the programming grade obtained by the participants may have skewed the generalisability of our findings. Since 72.7% of the participants in this study were male students, the findings in this study may be generalised to male students in introductory programming courses. Further, the number of participants who performed poorly in their introductory programming course was small (8.8%). Self-selection

bias is one factor which may have led to the small number of participants in the group that received a lower programming grade. Thus, the findings of this study may be generalised to students who obtained a good or excellent programming grade in their introductory programming course.

There was a strong positive relationship between programming self-efficacy and the emotional engagement factors. Enjoyment was clearly an important emotional engagement factor for success in programming. By contrast, the negative relationship between interest and programming performance did not support our initial hypothesis. Despite this, the negative relationship revealed interesting insights on how students perceived interest in their introductory programming course. Although the relationship between gratification and programming performance was not statistically significant, we argue that gratification has the potential to be an indicator of emotional engagement in introductory programming courses and could benefit from further research.

This study contributes to the design and delivery of introductory programming courses by making recommendations to introductory programming course instructors who wish to understand the behaviour of students and improve the performance of their students. Recommendations made include: to improve self-efficacy beliefs, embed enjoyable learning activities, and to develop the value-related interest of students. Additionally, this study revealed that the feeling of immediate gratification is an emotional engagement factor that appears to be unique to learning programming.

For future research, experiments could be carried out to examine the outcome of implementing the recommendations made to course instructors to improve the design and delivery of introductory programming courses. Studies could also be conducted in the next level of programming course that the student progress on to, in order to examine if the student's performance in the introductory programming course had an effect on their programming self-efficacy beliefs.

In conclusion, it is hoped that the findings and recommendations in this study would provide guidance to course instructors to develop and design courses that encourages introductory programming students to increase their self-efficacy beliefs and to be emotionally engaged with the course. In doing so, it is hoped that students may be able to improve their performance, thereby reducing the failure and attrition rates in introductory programming courses.

References

Akbulut, A. Y., & Looney, C. A. (2007). Inspiring students to pursue computing degrees. *Communications of the ACM, 50*(10), 67-71.

Appleton, J. J., Christenson, S. L., & Furlong, M. J. (2008). Student engagement with school: Critical conceptual and methodological issues of the construct. *Psychology in the Schools, 45*(5), 369- 386.

Appleton, J. J., Christenson, S. L., Kim, D., & Reschly, A. L. (2006). Measuring cognitive and psychological engagement: Validation of the Student Engagement Instrument. *Journal of School Psychology, 44*(5), 427-445.

Australian Government Department of Jobs and Small Business. (2018). *Australian Jobs 2018 is now available*. Retrieved October 15, 2018, from https://www.jobs.gov.au/news/australian-jobs-2018-now-available

Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review, 84*(2), 191-215.

Bandura, A. (1977a). *Social Learning Theory*. Englewood Cliffs, New Jersey: Prentice Hall.

Bandura, A. (1986). *Social Foundations of Thought and Action.* Englewood Cliffs, NJ: Prentice Hall.

Bandura, A. (1997). *Self-efficacy: The exercise of control.* New York: W H Freeman and Company.

Bartimote-Aufflick, K., Bridgeman, A., Walker, R., Sharma, M. & Smith, L. (2016). The study, evaluation, and improvement of university student self-efficacy. *Studies in Higher Education*, *41*(11), 1918-1942, https://doi.org/10.1080/03075079.2014.999319

Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming. *SIGCSE Bulletin, 39*(2), 32-36.

Betts, J. E., Appleton, J. J., Reschly, A. L., Christenson, S. L., & Huebner, E. S. (2010). A study of the factorial invariance of the Student Engagement Instrument (SEI): Results from middle and high school students. *School Psychology Quarterly*, *25*(2), 84-93.

Bishop-Clark, C., Courte, J., Evans, D., & Howard, E. V. (2007). A Quantitative and Qualitative Investigation of Using Alice Programming to Improve Confidence, Enjoyment and Achievement among Non-Majors. *Journal of Educational Computing Research, 37*(2), 193-207.

Boustedt, J., Eckerdal, A., McCartney, R., Mostrom, J. E., Ratcliffe, M., Sanders, K., & Zander, C. (2007). *Threshold concepts in computer science: do they exist and are they useful?* Paper presented at the Proceedings of the 38th SIGCSE technical symposium on Computer Science Education, Covington, Kentucky, USA. Retrieved from https://dl.acm.org/citation.cfm?doid=1227310.1227482

Bureau of Labour Statistics. (2018). *Occupational Outlook Handbook*. Retrieved October 15, 2018, from https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm

Bye, D., Pushkar, D., & Conway, M. (2007). Motivation, Interest, and Positive Affect in Traditional and Nontraditional Undergraduate Students. *Adult Education Quarterly, 57*(2), 141-158.

Chin, W. W. (1998). The partial least squares approach to structural equation modelling. In G. A. Marcoulides (Ed.), *Modern Methods for Business Research* (pp. 295–336). Mahwah, NJ: Lawrence Erlbaum Associates.

Christenson, S. L., Reschly, A. L., & Wylie, C. (2012). *Handbook of Research on Student Engagement*. Retrieved from http://VUW.eblib.com/patron/FullRecord.aspx?p=884351

Davis, W. A. (1982). A Causal Theory of Enjoyment. *Mind, 91*(362), 240-256, DOI: 10.2307/2253480

deBry, R. (2011). A learning space for beginning programming students. *Journal of Computing Sciences in Colleges, 27*(2), 4-11.

Ford, M., & Venema, S. (2010). Assessing the Success of an Introductory Programming Course. *Journal of Information Technology Education,* (9), 133-145.

Fornell, C., & Larcker, D. F. (1981). Evaluating Structural Equation Models with Unobservable Variables and Measurement Error. *Journal of Marketing Research, 18*(1), 39-50.

Fotaris, P., Mastoras, T., Leinfellner, R. & Rosunally, Y. (2016). Climbing up the Leaderboard: An Empirical Study of Applying Gamification Techniques to a Computer Programming Class. *Electronic Journal of e-Learning, 14*(2), 94-110.

Fredricks, J. A., Blumenfeld, P. C., & Paris, A. H. (2004). School Engagement: Potential of the Concept, State of the Evidence. *Review of Educational Research, 74*(1), 59-109.

Frenzel, A. C., Thrash, T. M., Pekrun, R., & Goetz, T. (2007). Achievement Emotions in Germany and China -- A Cross-Cultural Validation of the Academic Emotions Questionnaire-Mathematics. *Journal of Cross-Cultural Psychology, 38*(3), 302-309, DOI: http://dx.doi.org/10.1177/0022022107300276

Hair, J. F., Hult, G. T. M., Ringle, C. M., & Sarstedt, M. (2014). *A Primer on Partial Least Squares Structural Equation Modeling (PLS-SEM).* SAGE Publications, Inc.

Kanaparan, G., Cullen, R., & Mason, D. (2013). Self-Efficacy and Engagement as Predictors of Student Programming Performance. Paper presented at the *Proceedings of the 2013 Pacific Asia Conference on Information Systems* (PACIS 2013). Paper 282. https://aisel.aisnet.org/pacis2013/282

Kinnunen, P., & Simon, B. (2011). *CS majors' self-efficacy perceptions in CS1: results in light of social cognitive theory.* Paper presented at the Seventh International Workshop on Computing Education Research, Providence, Rhode Island, USA. Retrieved from https://dl.acm.org/citation.cfm?id=2016917

Krueger, R. A., & Casey, M. A. (2009). *Focus groups: A Practical Guide for Applied Research* (4th ed.). Thousand Oaks, California: SAGE Publications, Inc.

Liebenberg, J., Mentz, E., & Breed, B. (2012). Pair programming and secondary school girls' enjoyment of programming and the subject Information Technology (IT). *Computer Science Education,* 22(3), 219-236, DOI: 10.1080/08993408.2012.713180

Linnenbrink, E. A., & Pintrich, P. R. (2003). The Role of Self-efficacy Beliefs in Student Engagement and Learning in the Classroom. *Reading & Writing Quarterly, 19*(2), 119-137.

MacKenzie, S. B., Podsakoff, P. M., & Podsakoff, N. P. (2011). Construct measurement and validation procedures in MIS and behavioral research: Integrating new and existing techniques. *MIS Quarterly*, *35*(2), 293-334.

Maguire, P., Maguire, R., Hyland, P., & Marshall, P. (2014). Enhancing collaborative learning using pair programming: Who benefits? *AISHE-J: The All Ireland Journal of Teaching and Learning in Higher Education, 6*(2), 1411-14125.

McKinney, D., & Denton, L. F. (2004). Houston, we have a problem: there's a leak in the CS1 affective oxygen tank. *SIGCSE Bulletin,* 36(1), 236-239, DOI: 10.1145/1028174.971386

Mills, N., Pajares, F., & Herron, C. (2007). Self-efficacy of College Intermediate French Students: Relation to Achievement and Motivation. *Language Learning, 57*(3), 417-442, DOI: 10.1111/j.1467-9922.2007.00421.x

Mischel, W., Shoda, Y., & Rodriguez, M. (1989). Delay of gratification in children. *Science, 244*(4907), 933-938. DOI: 10.1126/science.2658056

Moore, G. C., & Benbasat, I. (1991). Development of an Instrument to Measure the Perceptions of Adopting an Information Technology Innovation. *Information Systems Research, 2*(3), 192-222.

O'Keefe, P. A., & Linnenbrink-Garcia, L. (2014). The role of interest in optimizing performance and self-regulation. *Journal of Experimental Social Psychology, 53*(0), 70-78, DOI: http://dx.doi.org/10.1016/j.jesp.2014.02.004

Pajares, F. (1997). Current Directions in self-efficacy research. In M.Maehr & P. R. Pintrich (Eds.), *Advances in Motivation and Achievement* (Vol. 10, pp. 1 - 49). Greenwich, CT: JAI Press.

Phan, H. P. (2011). Interrelations between self-efficacy and learning approaches: a developmental approach. *Educational Psychology, 31*(2), 225-246.

Podsakoff, P. M., MacKenzie, S. B., Lee, J. Y., & Podsakoff, N. P. (2003). Common Method Biases in Behavioral Research: A Critical Review of the Literature and Recommended Remedies. *Journal of Applied Psychology, 88*(5), 879 - 903.

Podsakoff, P. M., & Organ, D. W. (1986). Self-reports in Organizational Research: Problems and Prospects. *Journal of Management, 12*, 531-544.

Putwain, D., Sander, P., & Larkin, D. (2013). Academic self-efficacy in study-related skills and behaviours: Relations with learning-related emotions and academic success. *British Journal of Educational Psychology, 83*(4), 633-650. DOI: 10.1111/j.2044-8279.2012.02084.x

Quille, K., & Bergin, S. (2016). *Programming: Further Factors That Influence Success.* Paper presented at the Twenty-seventh Annual Workshop of the Psychology of Programming Interest Group (PPIG 2016), Cambridge, UK. Retrieved from http://www.ppig.org/library/paper/programming-further-factors-influence-success

Ramalingam, V., LaBelle, D., & Wiedenbeck, S. (2004). Self-efficacy and mental models in learning to program. *SIGCSE Bulletin, 36*(3), 171-175.

Ramalingam, V., & Wiedenbeck, S. (1998). Development and Validation of Scores on a Computer Programming Self-Efficacy Scale and Group Analyses of Novice Programmer Self-Efficacy. *Journal of Educational Computing Research, 19*(4), 367 - 381.

Reschly, A. L., & Christenson, S. L. (2012). Jingle, Jangle, and Conceptual Haziness: Evolution and Future Directions of the Engagement Construct. In S. L. Christenson, A. L. Reschly & C. R. F. Wylie (Eds.), Handbook of Research on Student Engagement. Dordrecht: Springer Science+Business Media New York. Retrieved from http://VUW.eblib.com/patron/FullRecord.aspx?p=884351.

Ringle, C. M., Wende, S., & Becker, J.-M. (2015). *SmartPLS 3.* Boenningstedt: SmartPLS GmbH, http://www.smartpls.com.

Schunk, D. H. (1996). *Self-Efficacy for Learning and Performance.* Paper presented at the Annual Meeting of the American Educational Research Association, New York. Retrieved from https://files.eric.ed.gov/fulltext/ED394663.pdf

Schunk, D. H., & Mullen, C. A. (2012). Self-Efficacy as an Engaged Learner. In S. L. Christenson, A. L. Reschly & C. Wylie (Eds.), *Handbook of Research on Student Engagement*. Dordrecht: Springer Science+Business Media New York. Retrieved from http://VUW.eblib.com/patron/FullRecord.aspx?p=884351.Silvia, P. J. 2003. "Self-efficacy and interest: Experimental studies of optimal incompetence," *Journal of Vocational Behavior* (62:2), pp 237-249, DOI: http://dx.doi.org/10.1016/S0001-8791(02)00013-1

Sheard, J., Carbone, A., & Hurst, A. J. (2010). Student engagement in first year of an ICT degree: staff and student perceptions. *Computer Science Education, 20*(1), 1-16, DOI: 10.1080/08993400903484396

Sherer, M., Maddux, J. E., Mercandante, B., Prentice-Dunn, S., Jacobs, B., & Rogers, R. W. (1982). The Self-Efficacy Scale: Construction and Validation. *Psychological Reports*, *51*(2), 663-671. DOI: 10.2466/pr0.1982.51.2.663

Sorva, J. (2010). *Reflections on Threshold Concepts in Computer Programming and Beyond.* Paper presented at the Proceedings of the 10th Koli Calling International Conference on Computing Education Research, Koli, Finland. Retrieved from https://dl.acm.org/citation.cfm?doid=1930464.1930467

Stangor, C. (2011). *Research Methods for the Behavioral Sciences*. Belmont, CA, USA: Wadsworth Cengage Learning.

Thomas, D. R. (2006). A General Inductive Approach for Analyzing Qualitative Evaluation Data. *American Journal of Evaluation, 27*(2), 237-246. DOI: 10.1177/1098214005283748

Walker, C. O., Greene, B. A., & Mansell, R. A. (2006). Identification with academics, intrinsic/extrinsic motivation, and self-efficacy as predictors of cognitive engagement. *Learning and Individual Differences*, *16*(1), 1-12. https://doi.org/10.1016/j.lindif.2005.06.004

Watson, C., Li, F. W. B., & Godwin, J. L. (2014). *No tests required: comparing traditional and dynamic predictors of programming success.* Paper presented at the Proceedings of the 45th ACM Technical Symposium on Computer Science Education, Atlanta, Georgia, USA. https://dl.acm.org/citation.cfm?doid=2538862.2538930

Wiedenbeck, S., Xiaoning, S., & Chintakovid, T. (2007). Antecedents to End Users' Success in Learning to Program in an Introductory Programming Course. *Paper presented at the IEEE Symposium on Visual Languages and Human-Centric Computing, 2007.* VL/HCC 2007. Retrieved from http://ieeexplore.ieee.org/document/4351328/

Yazzie-Mintz, E., & McCormick, K. (2012). Finding the Humanity in the Data: Understanding, Measuring and Strengthening Student Engagement. In S. L. Christenson, A. L. Reschly & C. Wylie (Eds.), *Handbook of Research on Student Engagement* (pp. 743-762). Dordrecht: Springer Science+Business Media New York. Retrieved from http://VUW.eblib.com/patron/FullRecord.aspx?p=884351.Yip, M. C. W. 2012. "Learning strategies and self-efficacy as predictors of academic performance: a preliminary study," *Quality in Higher Education* (18:1), pp 23-34. DOI: 10.1080/13538322.2012.667263

Yip, M. C. W. (2012). Learning strategies and self-efficacy as predictors of academic performance: a preliminary study. *Quality in Higher Education*, *18*(1), 23-34. https://doi.org/10.1080/13538322.2012.667263

Zimmerman, B. J. (2000). Self-Efficacy: An Essential Motive to Learn. *Contemporary Educational Psychology, 25*(1), 82-91.

Zingaro, D. (2014). *Peer instruction contributes to self-efficacy in CS1.* Paper presented at the 45th ACM technical symposium on Computer Science Education, Atlanta, Georgia, USA. https://dl.acm.org/citation.cfm?doid=2538862.2538878

Appendices

Appendix A: Survey Items

Construct:	**Programming self-efficacy**	**Item adapted from**
Factor	**Item**	
Note: All items start with *"I am confident that……."*		
General self-efficacy	I can complete a new task that is assigned to me if I keep trying.	Bresó et al. (2011)
	I can achieve the goals that I set for myself in this programming course.	
	I have the capability to learn the contents of this programming course.	
	When I learn something new, I will not give up easily if I am not successful initially.	Sherer at al. (1982)
	When unexpected problems occur I can handle them well.	
	I can stick to completing a task even though it may be unpleasant.	
Independence and persistence	I can complete my programming assessment if I had a lot of time.	Ramalingam & Wiedenbeck (1998)
	I can complete my programming project if I refer to resources such as the built-in help, programming reference manuals, or online programming forums.	
	I can find ways of overcoming the problem if I get stuck at a point while working on a programming assessment.	
	I can correct (debug) all the errors in a program that I have written, and make it work.	
Complex programming tasks	I can apply the right programming concepts to solve a problem given to me.	
	I can understand and apply the fundamental object-oriented programming concepts used in my programming course.	
	I can make use of a pre-written library in the programming integrated development environment (IDE).	
	I can write a program to solve any given problem as long as the specifications are clear.	
	I can mentally trace through the execution of a complex program given to me.	
	I can find a way to concentrate on my programming project, even when there are many distractions around me.	
	I can find ways of motivating myself to program, even if the problem area was of no interest to me.	
Simple programming tasks	I can write programming code that runs without errors (no syntax errors).	
	I can write programming code that is logical.	
	I can write a small program for a small problem that is familiar to me.	

Construct:	**Programming self-efficacy**	**Item adapted from**
Factor	**Item**	
Note: All items start with *"I am confident that……."*		
	I can understand the language structure and the usage of the reserved words /keywords in the programming language.	
	I can write a reasonably sized program that can solve a problem that is only vaguely familiar to me.	

Construct:	**Enjoyment**	**Item adapted from**
Item		
My programming course is stimulating compared to the other courses that I am enrolled in.		Bishop-Clark et. al (2007)
I like writing programs.		
The challenge of coding solutions to programming problems appeals to me.		

Construct:	**Interest**	**Item adapted from**
Item		
My programming assessments take priority over the assessments from other courses.		Focus groups
I am passionate about programming.		
My programming course suits me better than the other courses that I am currently enrolled in.		

Construct:	**Gratification**	**Item adapted from**
Item		
I feel a deep sense of satisfaction when I finally get my program to work.		Focus groups
I feel relieved when I complete my programming assessment.		
I feel a strong sense of achievement when I complete my programming assessments.		
Once I complete my programming assessment, I feel pleased that I have successfully completed a challenging piece of assessment.