PRIMM Model Towards Malaysian Matriculation Students' Motivation in Learning Programming

Tevya Letchumanan, Sarawak Matriculation College, Malaysia

The Asian Conference on Education 2024
Official Conference Proceedings

Abstract

As computer programming becomes one of the most sought-after skills, novice learners, especially Malaysian Matriculation students, struggle to grasp the ideas of computational thinking. The lack of time in the traditional classroom also inhibits constructive communication. These also cause high stress and low self-efficacy in learning programming languages. This research has been guided by constructivism to design a Predict-Run-Investigate-Modify-Make (PRIMM) module that will allow learners to think at a higher-order level via social constructivism theory, engagement theory, and self-determination theory. The literature has strengthened the role of PRIMM in improving the academic performance of students. However, there is a gap in how the PRIMM model improves motivation, engagement, and academic success among pre-university students in the Malaysian context. This research aims to design a PRIMM module by underpinning the constructivism framework, exploring the role of the PRIMM model in mediating learning motivation, authentic engagement, and reducing stress, and validating the module via a focus group. Phenomenological research was carried out to achieve these objectives, and volunteer participants were subjected to focus group interviews. Five major themes emerged from the focus group interview: Motivated, Engaged, Enjoyable, Efficient, and Collaboration. Research findings state that the PRIMM module developed has improved collaboration, learning interest, learning motivation, and authentic engagement and has reduced stress in learning programming languages among beginners. The findings of this research provide a framework for lecturers to redesign the computer science classrooms in Malaysian matriculation colleges via classroom activities that implement problem-solving before formal instructions.

Keywords: PRIMM Model, Novice Programmers, Malaysian Matriculation

Introduction

Background of Study

Computational thinking skills such as problem-solving, recursive thinking, and programming knowledge are no longer considered skills exclusively required for computer scientists but for professionals from the engineering, economics, and education analytical fields (Apiola et al., 2010; Song et al., 2021). For novice students, however, the introduction to computational thinking, syntaxes, semantics, and overall structure of programming languages are all complex and overwhelming to learn over a short period. Even after learning the theories, students struggle to apply the content they have learned to build a robust program, which often leads to frustrations and a loss of interest and motivation, resulting in a higher drop rate. It is essential to continuously maintain students' motivation, engagement, passion, and joy to ensure that students successfully master programming. Motivation is often associated with learning via self-regulation, metacognition, and persistence. One way to achieve this is via the use of the PRIMM model. The Predict-Run-Investigate-Modify-Make (PRIMM) approach was built based on the Use-Modify-Create (UMC) model, where students read and predict what a program will do before investigating the actual output and finally build their code based on the comprehension from previous steps. Since student-centered learning has been encouraged in educational institutions to improve student accountability, student engagement, and active learning, the PRIMM model can be implemented to meet the demands of the knowledge and skills learners are expected to obtain at the end of each lesson. Specifically, the approach requires students to predict an output based on prior knowledge or logic, allowing students to learn from the success or failure to guess the outputs and providing them more time to engage in the classroom.

Problem Statements

Students face a lack of constructive communication between peers and the instructor (Garvin-Doxas & Barker, 2004). In the conventional classroom, teachers spend a substantial amount of time delivering the subject matter, resulting in less discussion time. The lack of communication results in students being unable to build knowledge to solve problems and failing to get guidance to solve the problems they could potentially have. Consequently, students also tend not to get any support to fix their misconceptions. This goes in line with Bandura's social cognitive theory that emphasises the occurrence of learning in a social context. While the study by Sentance and colleagues (2019a) on the PRIMM model itself has reported that students taught via the PRIMM model have academically performed better than students in the traditional control group, there is a gap in studies on the implication of the PRIMM model amongst Malaysian Matriculation Students via constructive communication that can improve motivation given the 64 hours allocated per semester to teach computer science, specifically Java Programming in Malaysian Matriculation Program.

Another problem that learners face when learning a programming language is the lack of problem-solving skills in computer science (Cheah, 2020). While students can understand the syntax theoretically, they fail to build high-level analytical thinking that can produce effective solutions to a problem causing a lack of motivation to learn programming. Additionally, applying programming knowledge in different contexts has become a struggle because students tend only to have surface knowledge of content-specific programming. Studies on the UMC model, such as those by Martin and colleagues (2020) and Salac and fellows (2020), have indicated the role of the model in improving computational thinking.

However, there is a gap in studies with these two elements overlapping in teaching programming and a gap in the studies on the role of motivation in improving engagement when studying computer programming. There is also a gap in the studies for these elements with the PRIMM model, especially for the Malaysian Matriculation setting.

Learners have also reported stress in learning computer programming due to low self-efficacy (Rahardjo et al., 2013). According to Figueiredo and Garcia-Penalvo Jr (2020) and Amoloza (2015), independently working on programming subjects leads students to a higher cognitive load. This induces stress, which has a detrimental impact on engaging with lessons and, thus, learning programming as a whole. This negatively affects the retention rate and lowers the students' achievement rate. In addition, learners tend to think of computer science tasks and problems as harder than they actually are. This leads to constant self-criticism, and learners stop engaging with the classroom lessons and activities without any motivation to learn. Low self-efficacy also limits the learners' positive attitudes toward learning programming languages and their cognitive skills. Although studies by Piteira and Costa (2013) and Salguero and colleagues (2021) have pointed out the stress and anxiety faced by students when learning programming, there is a lack of research on how the issue can be approached via the learning engagement caused by the PRIMM model.

Research Questions

1. How to design and develop teaching and learning module based on the PRIMM model for Malaysian matriculation computer science students?
2. What is the role of the PRIMM model in mediating learning motivation when learning computer programming?
3. How does learning motivation improve students' authentic engagement when learning Java programming through the PRIMM model?
4. What is the role of the PRIMM model in reducing the stress faced by students when learning computer programming?

Purpose and Objectives

In response to the challenges faced by novices in learning Java programming, the purpose of the study is to examine the implementation of the PRIMM model in improving Malaysian matriculation computer science students' motivation to learn computer programming. The objectives of the research are:

Main Objective: To design and develop teaching and learning module based on the PRIMM model for Malaysian matriculation computer science students by applying the constructivism elements.

Specific Objectives:

1. To explore the role of the PRIMM model in mediating the learning motivation when learning computer programming.
2. To discover the role of motivation in mediating students' authentic engagement when learning computer programming through the PRIMM model.
3. To understand the role of the PRIMM model in reducing the stress faced by students when learning computer programming.

Literature Review

PRIMM Model

The PRIMM model concentrates on having students discuss how and why computer programs operate before the students edit and write their programs Sentance and colleagues (2019b). This method addresses the issue of beginners developing programs before they are yet able to read them. The model primarily draws from three main areas of research, namely, Tracing and read-before-you-write by Lister and colleagues (2009a), Use-Modify-Create by Lee and colleagues (2011), and Levels of Abstraction by Perrenet and colleagues (2005a). PRIMM encourages self-regulation and metacognition while providing students and teachers with a language to discuss the tactics they are employing. According to Sentence (2021), the PRIMM model focuses on the five principles: read code before you write code, work collaboratively to talk about programs, focus on code comprehension, use existing starter programs, and gradually take ownership of programs. The five stages of the PRIMM are explained based on the study by Sentence and Waite (2017), as shown in Figure 1.

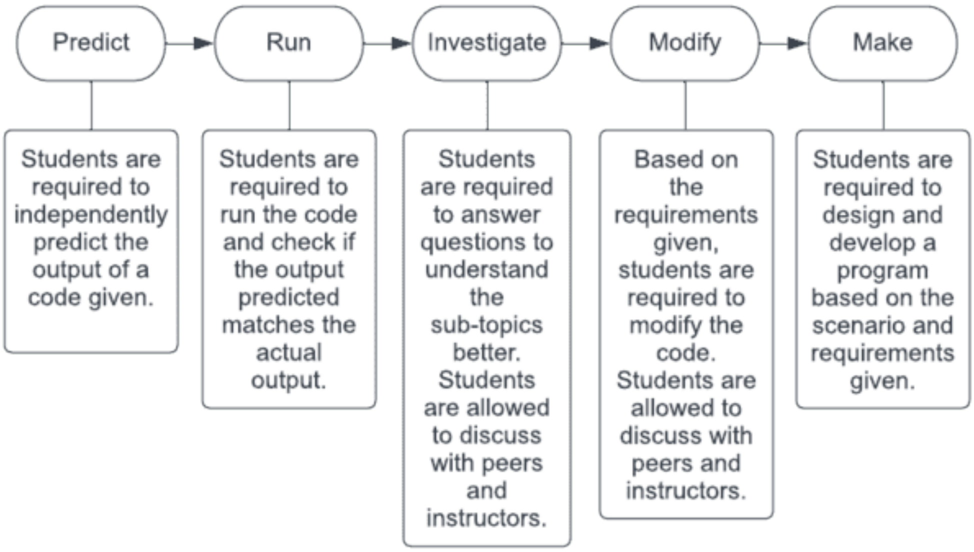

Figure 1: Five Stages of the PRIMM Model

Malaysian Matriculation Programme

The Malaysian Matriculation Programme is a pre-university programme with two and four-semester programs. Learners who have completed Sijil Pelajaran Malaysia (SPM) and are not over 20 years old are eligible to apply for the programme has met the grade requirements. Upon completing the program, learners pursue their degree within local public universities, private educational institutions, or international institutions. Science and accounting streams are available within the two-semester system (SDS), and the four-semester system (SES) offers science, accounting, and engineering streams. The Science stream offers modules for Life Science, Physical Science, and Computer Science. The computer science subject is taught as part of the Physical Science and Computer Science modules. After the first two semesters, the SES students will follow the exact curriculum specifications drafted for SDS. The Curriculum Specification (CS) for the SDS semester 2 Computer Science (SC025) used in this research is attached in Appendix A.

Learning Motivation

Learning motivation acts as the driving force for students to achieve learning goals and to strengthen and improve knowledge acquisition Shabani (2016). It is an essential predictor of the learning success rate (Schnotz et al., 2009). At the initial and bottleneck stages of learning, extrinsic motivation yields better results until learning becomes autonomous, after which students gain fun and a sense of achievement via intrinsic motivation (Lin et al., 2017).

Motivation on the students' side is crucial for successful cumulative learning due to the importance of persistence when learning new concepts. Learning motivation in a classroom implementing the PRIMM model reflects students' engagement and contribution during all five stages of the model. To ensure the effectiveness of the PRIMM model, students are expected to be actively engaged in all class activities, and for this, students must be motivated to participate first. Despite intrinsic motivation being stated to be superior, most human endeavours cannot be intrinsically motivated after childhood because the ability to be intrinsically driven is curtailed by societal expectations. Therefore, in this study, intrinsic and extrinsic motivation are both explored to measure the dimensions of learning motivation in a programming classroom taught using the PRIMM model.

Learning Engagement

Student learning engagement is the involvement of students in their studies and the persistence that the students must have to accomplish desired goals (Saeed & Zyngier, 2012). Students are said to be engaged when they are involved, committed, and attentive in classroom activities despite the obstacles faced in completing the tasks. Motivation and academic engagement are reciprocal, as motivation may affect students' engagement in academic tasks (Singh et al., 2002; Z. Wu, 2019). The levels of engagement are examined based on the five dimensions: authentic, engagement, passive compliance and rebellion (Digamon & Cinches, 2017). In authentic engagement, students have a high level of curiosity and personal significance in their work and do not back down from a difficulty. Saeed and Zyngier (2012) propose an analysis of the type of motivation that has a stronger association with authentic student engagement since students' intrinsic motivation showed authentic engagement while students with extrinsic motivation only showed ritual engagement that may not fully assist learning. As such, this study focuses on the role of intrinsic and extrinsic motivation in mediating the students' authentic engagement when the PRIMM model is implemented in the classroom to teach programming.

Stress in Learning

Stress is when an individual finds a circumstance or challenge that surpasses their ability to solve it (Lindau et al., 2016). As such, the interpretation of the stress level is subjective for each student based on their cognitive capacity. The perception of stress in learning is not solely dependent on the volume of learning tasks but can also be influenced by the nature of the learning expected (Rudland et al., 2020). Challenges can manifest in various forms, including grappling with complex subject matter, mastering a particular skill, or deciphering critical information from intricate patient histories. Both learners and educators have the potential to shape the intensity and nature of the stressors encountered, although often, it is the educator who determines these factors, in addition to the characteristics of the learning environment. The study by Von Hausswolff and colleagues (2020a) states that the decrease in stress associated with hands-on learning could explain the small positive effect on the test after one week. The connection between cognition and bodily feeling is highlighted, suggesting that emotions play a role in understanding and meaning-making. Therefore, managing students' stress levels is crucial to maximise learning motivation and information retention.

Guided Theory

The study is guided by three theories namely, self-determination theory, engagement theory, and social constructivism theory. These theories are used to support the study in identifying the role of the PRIMM model in improving learning motivation programming skills and reducing stress. Figure 2 shows the incorporation of the theories into developing the PRIMM module.

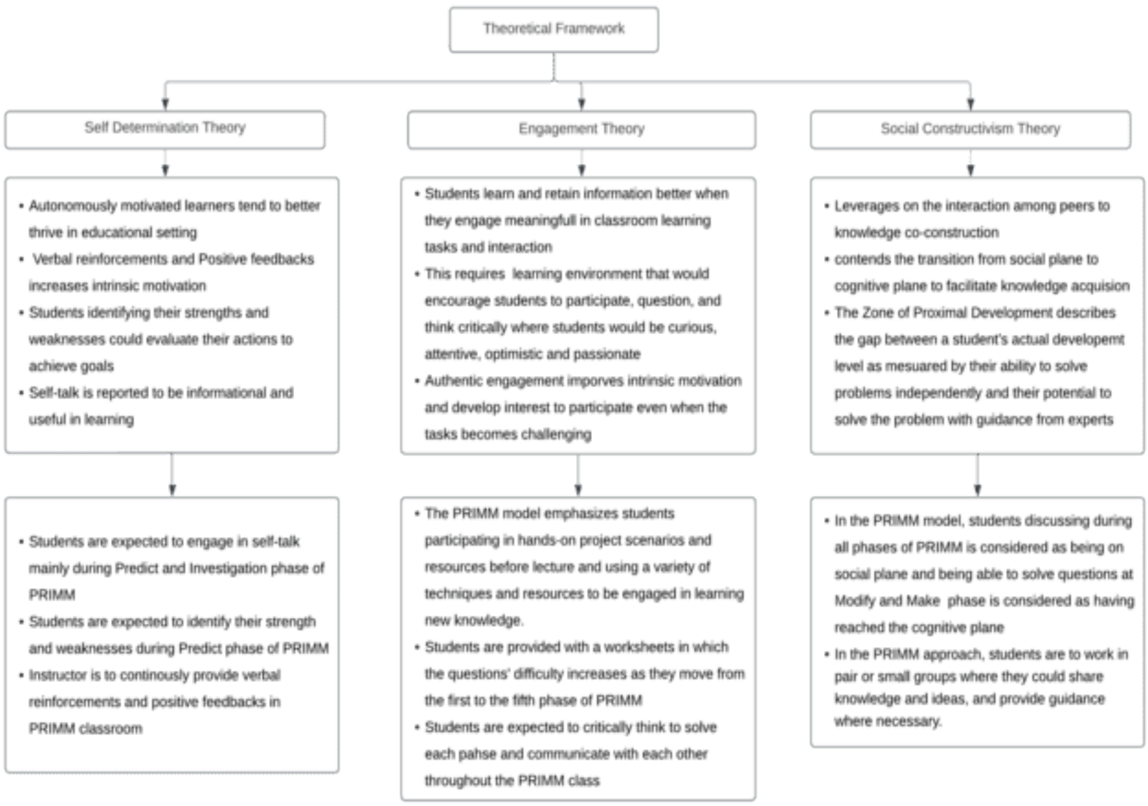

Figure 2: The Guided Theories in Developing the PRIMM Module

Method

Research Design

A phenomenological research design was used as part of the qualitative research approach. The phenomenological design investigates an individual's lived experiences while barring the researcher's preconceived assumptions regarding the topic. In this research, the phenomenological design has been utilised to elicit the lived experience of novice learners who have experienced the PRIMM model to learn computer science. The selection of a phenomenological design enables an in-depth exploration of learners' motivation in learning computer science via the PRIMM model as well as the best techniques of the PRIMM model encountered in the implementation process. Designing a PRIMM module that is best suited for novice learners learning Java programming is also important.

The syllabus of the Matriculation computer science program is divided into four chapters. In the first chapter, Introduction to Programming, students learn about the programming language paradigm and the translator programs. In chapter two, Approach in Problem-Solving, students are taught the five problem-solving steps. In the third and fourth chapters, students learn about the identifiers, variables, constants, expressions, control structures, arrays, and methods of Java Programming Language.

The PRIMM model will be used to teach Chapter Three: Design a Solution involving the introduction to control structures and Chapter Four: Java Language, which includes sub-topics on data types, variables, constants, two-dimensional arrays, and methods. For each subtopic, students must complete a worksheet attached to the Google Classroom. Together, these six worksheets are to form an efficient module to teach SC025 in the Malaysian matriculation program. A total of 10 hours would be spent on each subtopic. While students complete the worksheets, the instructor provides explanations, scaffolding, and guidance toward a better understanding the topic. Verbal reinforcements via positive feedback would be given throughout the classes to leverage self-determination theory. As the students move towards the Modify and Make sections of the worksheets, guidance will be gradually withheld so that students can apply what they have learned. Students are expected to be intrinsically motivated to solve these questions. In each worksheet, five questions are set, each representing a phase of the PRIMM model. Six worksheets have been prepared to cover nine sub-topics. Table 1 in Appendix B shows the worksheet, the subtopics covered in the worksheets, and the learning objectives for each worksheet. The complete module, which consists of Worksheet 1 to Worksheet 6, is attached in Appendix C.

Sample and Sampling Method

The learners are all to be from the science stream, computer science subject in Sarawak Matriculation College to ensure all the participants have experienced the PRIMM classroom directly via the researcher. The context is Computer Science learners in the class for whom it is their first time learning programming for public examination. In this study, 24 learners who were registered in the computer science subject will be recruited. The learners were taught using (Bybee, 2009) 5E instructional model for the first two chapters and the newly developed PRIMM module for chapters three and four. A non-probability sampling was used because the study was qualitative. A purposeful sampling method was used to select study groups. In this context, criterion sampling has been used. In selecting the participants, criteria such as being voluntary, having not studied any programming language for exam purposes prior to entering the Matriculation program, and having attended at least 90% of the classes throughout the semester were sought. Plummer-D'Amato (2013) states that using four to five focus groups is adequate. As such, four focus groups were formed for this study. Six participants were recruited into each focus group to be interviewed for a maximum of 60 minutes as suggested by Giibs (1997). To bar any gender bias and to ensure comfortable focus group interviews, each group was formed to have three male and three female students.

Data Collection Procedures

The data collection process for this study is done after students have been introduced to the complete module via focus group interviews. The data was collected from the participants who have learned Java programming language via the PRIMM module over 16 weeks. Figure 3 shows the flow and timeline of module development and data collection.

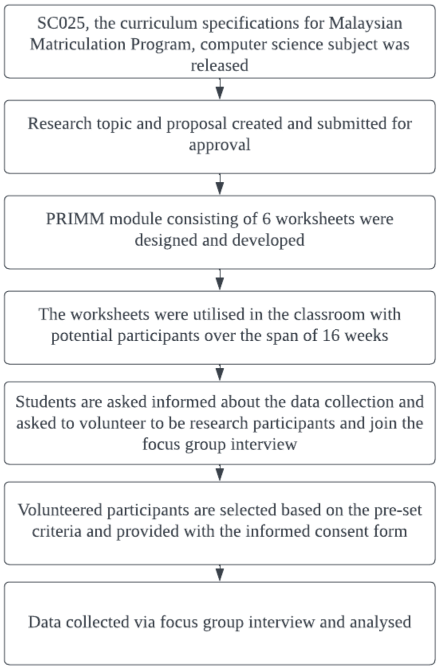

Figure 3: Flow and Timeline of Module Development and Data Collection

Focus group interviews were conducted with the learners to analyse the design of the learning module based on PRIMM and the effectiveness of the new PRIMM module. The focus group interviews have been used to discuss different perspectives on the PRIMM module approach within a limited duration. Based on the study conducted by Fern (1982), Plummer-D'Amato (2013), and Rabiee (2004), the participants were grouped into three groups of six for focus group interviews with a maximum time duration of 60 minutes so that enough input could be derived from the discussion without exhausting the participants. Each interview was conducted online, using Google Meet. The interview was recorded with the participant's permission and stored in Google Drive with a password.

According to Bevan (2014), phenomenological interviewing should focus on collecting data to examine participants' lived experiences. Therefore, in-depth information concerning participants' experiences must be gathered via open questions. The interview protocol for this study starts with asking participants to describe the PRIMM classroom they participated in in detail. Based on the participants' answers, they were questioned on the techniques used to complete the PRIMM worksheets and how they helped them. To answer the research questions, participants were questioned on how the newly developed PRIMM module has helped them to improve communication, learning motivation, and engagement. The interview protocol for the focus group interview with learners is attached in Appendix D.

Data Analysis Procedures

Each focus group interview was immediately transcribed with behavior annotations and phonetic transcription of dialects and filler words. The interview transcription is attached in Appendix E. The transcriptions include annotations for behavior to assist in analysing learners' emotions concerning learners' motivation state as a result of participating in the PRIMM module. Phonetic transcription of dialects and filler words is included since the research population comprises learners from Sarawak Matriculation College who mixed the local dialect during the focus group interview.

The focus group interviews were analyzed using thematic analysis as this provided flexibility and a comprehensive explanation of data. Thematic analysis was useful when looking for subjective information about learners' experiences and views of the PRIMM model and the newly developed module. Themes were identified using inductive coding to free the research from the researcher's theoretical interest. The researcher then grouped the recurring topics during the data collection and grouped the codes into themes to summarize the data sections in a useful way to achieve research objectives. Next, the researcher looked for common codes to form themes relevant to the research questions, reviewed the selected themes, and then finalized the themes. These steps must be done sequentially to ensure the data are relevant to answer the research questions.

Qualitative data coding is used to create and assign codes to categorize data extracts. The codes are then used to derive themes and patterns for thematic analysis. At this stage, similar data types were labeled and grouped to generate themes and ensure that data analysis was manageable. Inductive coding was used for this research. As such, the researcher developed the codes based on what was found in the data. Inductive coding has been used since the data collection involves interviews, and the direction of the conversation has yet to be known. Inductive coding was also selected for this research since there exists a big gap in research on the PRIMM classroom among pre-university computer science learners.

In the first stage, broad codes were derived from the interview, and structural coding was used to describe the data in a condensed manner. The data is then coded based on the interview transcription so that data can be structured into smaller pieces for further analysis. The researcher then coded line-by-line to add details to the codes. The researcher used the major themes and subthemes for the data to be synthesized. The researcher went through all the transcripts and the codes to conduct quality appraisals and analyze the data and its quality to ensure the accuracy of data for each code.

A total of 24 students participated in four focus group interviews whereby 50% of the participants were male and the other female. In this paper, participants were identified by group number and random participant number within that group, and participants' quotations are indicated in italics. Quotations in languages other than English have been typed in red font and a translation has been provided in parentheses.

Discussion

The complete version of the themes, subthemes, and participant quotes are attached in Appendix F.

The main idea of collaboration that the PRIMM module has allowed is the ability to discuss not only each other's answers but also each other's ideas. Participants pointed out that they could see multiple algorithms for the same question and test each one to find the most optimal solution. Discussions were sometimes also made with the instructor for further understanding. It has also allowed students to study the problem as a group and deduce ideas about each subtopic. Therefore, when completing the worksheets, students learn about the concepts with similar-minded peers. In this situation, students learn and teach their peers, which, according to the participants, has helped them to remember the concepts better. The findings in accordance with the research questions are discussed in the paragraphs below.

Research Question 1: How to design and develop a teaching and learning module based on the PRIMM model for Malaysian matriculation computer science students?

To address the main objective of the research and the first research question, the design of a PRIMM module to teach programming in the Malaysian Matriculation Program is best when it has scope for students to practice coding in the classroom with peer communication and collaboration as well as a step-by-step increase in terms of difficulty with a relationship between the worksheets such that each worksheet can be used as a reference. The findings also suggest that each worksheet can have more than one question for each phase of PRIMM to allow better exposure. The findings regarding the provision of the module's answer scheme have contradicted. Some participants have found that not having an answer scheme is more motivating since they do not have to focus on getting the exact-worded, correct answers. However, participants have also stated that having an answer scheme would further increase the probability of using the module as revision material. A study by Sinta and colleagues (2019) on students' improvement in learning grammar via Quizziz has found that releasing the answer key to the students improved their ability to identify their strengths and weaknesses, improving their learning. Therefore, it is suggested that an answer scheme is prepared and given to students after the worksheet in the module has been completed.

Research Question 2: What is the role of the PRIMM model in mediating learning motivation when learning computer programming?

This PRIMM model has developed intrinsic motivation amongst the students, where students tend to continue doing the worksheets despite getting the answers wrong in the first round to achieve personal satisfaction. Theories and applications of intrinsic motivation are connected to fostering respectful exchanges and facilitating profound learning. This is partly attributed to individuals channelling their efforts into activities that naturally provide a sense of fulfilment (Ginsberg & Wlodkowski, 2019). The module has also extrinsically motivated students via healthy competition. It has been found that students are motivated to learn and complete the module after seeing their peers complete it. Participants stated that seeing their peers understand the concept and solve the questions encouraged them to continue to learn even when they had gotten the prediction wrong or struggled to solve the questions in the previous worksheets. On the other hand, once they start getting better at solving the questions, the feeling of satisfaction eventually improves learning interest. The module also reassures students that with practice, they will be able to master the content.

Research Question 3: How does learning motivation improve students' authentic engagement when learning Java programming through the PRIMM model?

Learning motivation has been a stepping stone in improving authentic engagement in PRIMM classrooms, consistent with the literature review. The direct impacts of intrinsic or extrinsic motivation on academic performance have been evident through learning engagement (H. Wu et al., 2020). The PRIMM module developed has fostered authentic engagement through high-level curiosity, personal significance, and determination. The engagement was also observed when students actively discussed the questions and answers with each other. All participants stated and agreed that the motivation they developed to complete the module to get the satisfaction of completing it kept them engaged in the classroom. It has also been found that students have also run the code from Question multiple times after checking their prediction in an attempt to understand the concepts better before attempting Question 3.

Research Question 4: What is the role of the PRIMM model in improving Java programming skills among Malaysian matriculation computer science students?

It has been found that solving all questions correctly without the instructor's intervention during the first worksheet was nearly impossible for all participants. However, participants were able to solve the worksheets, especially Question 5, on either the first attempt or after a short discussion with their peers. This assures that continuous hands-on practice is a form of higher cognitive work as supported by Zainuddin and Halili (2016) which has improved the matriculation students' Java programming skills. The study also found that the PRIMM module paved the way for understanding programming language concepts by doing the programming itself. Understanding the concepts and the curiosity to learn resulted in deeper learning. The Module has also allowed students to detect their weaknesses and learn from their mistakes. This way, students made fewer errors as they progressed through the worksheets and developed better algorithms to solve Question 5 faster. This learning method is also supported by Fischer and colleagues (2006). According to the authors, deriving lessons from mistakes and close calls and individual and systemic accountability are important in promoting knowledge acquisition.

Research Question 5: What is the role of the PRIMM model in reducing the stress faced by students when learning computer programming?

The findings suggest that the PRIMM model improved learning interest and reduced students' learning stress. For students who have continuously heard that programming is difficult and believed in the perception, completing the model has given them the confidence to believe in their abilities. The discussions among peers have tremendously helped when students were stuck with solving the questions and fixing the bugs which alleviated students' stress and is in line with the study conducted by Choi and colleagues (2021). For students who struggled to convert their theoretical knowledge into the application, implementing the module helped them gain hands-on experience in coding while learning the concepts.

Conclusion

The review of literature has exhibited success on the PRIMM model in improving learning performance and at the same time, has pointed out the gap in the mediating role of learning motivation towards academic performance, especially in the scope of the Malaysian matriculation program. There is a huge gap in the implementation of the PRIMM model among pre-university computer science students, especially in the Malaysian context. While studies have been conducted on the success of the PRIMM model towards the academic performance of learners, the impact of the model on motivation and engagement, as well as the impact of the model in pre-university and Malaysian student settings, is still unknown. Underpinning the engagement theory, social constructivism, and self-determination theory, this research aims to design and develop a PRIMM module suitable for computer science learners of the Malaysian matriculation program. Research findings from the focus group interviews and thematic analysis found that the PRIMM module was efficient in learning programming, and students were motivated to learn and enjoyed the PRIMM classroom, where students had the opportunity to collaborate with peers and were engaged throughout the classroom. The instructors and learners must have a say in understanding the perspective of teaching and learning. This research could act as a guide for instructors who are planning on redesigning the computer science classrooms for lecturers from matriculation colleges around Malaysia to develop new in-class activities. The PRIMM module that was developed can be efficiently used by Malaysian Matriculation colleges to teach programming.

References

Apiola, M., Lattu, M., & Pasanen, T. A. (2010). Creativity and intrinsic motivation in computer science education: Experimenting with robots. *ITiCSE'10 - Proceedings of the 2010 ACM SIGCSE Annual Conference on Innovation and Technology in Computer Science Education*, 199–203. https://doi.org/10.1145/1822090.1822147

Bevan, M. T. (2014). A Method of Phenomenological Interviewing. *Http://Dx.Doi.Org/10.1177/1049732313519710*, *24*(1), 136–144. https://doi.org/10.1177/1049732313519710

Bybee, R. W. (2009). THE BSCS 5E INSTRUCTIONAL MODEL AND 21ST CENTURY SKILl. *The National Academies Board on Science Education*, 2–24. https://sites.nationalacademies.org/cs/groups/dbassesite/documents/webpage/dbasse_073327.pdf

Cheah, C. S. (2020). Factors contributing to the difficulties in teaching and learning of computer programming: A literature review. *Contemporary Educational Technology*, *12*(2), 1–14. https://doi.org/10.30935/CEDTECH/8247

Choi, J. A., Kim, O., Park, S., Lim, H., & Kim, J. H. (2021). The Effectiveness of Peer Learning in Undergraduate Nursing Students: A Meta-Analysis. *Clinical Simulation in Nursing*, *50*, 92–101. https://doi.org/10.1016/j.ecns.2020.09.002

Digamon, J. S., & Cinches, Ma. F. C. C. (2017). Schlechty's student engagement continuum in the work team experience: A pilot study. *Journal of Institutional Research South East Asia*, *15*(3), 5–18. https://www.researchgate.net/publication/321921173_Schlechty's_student_engagement_continuum_in_the_work_team_experience_A_pilot_study

Fern, E. F. (1982). The Use of Focus Groups for Idea Generation: The Effects of Group Size, Acquaintanceship, and Moderator on Response Quantity and Quality. *Journal of Marketing Research*, *19*(1), 1. https://doi.org/10.2307/3151525

Figueiredo, J., & Garcia-Penalvo, F. J. (2020). Increasing student motivation in computer programming with gamification. *IEEE Global Engineering Education Conference, EDUCON*, *2020-April*, 997–1000. https://doi.org/10.1109/EDUCON45650.2020.9125283

Fischer, M. A., Mazor, K. M., Baril, J., Alper, E., DeMarco, D., & Pugnaire, M. (2006). Learning from mistakes: Factors that influence how students and residents learn from medical errors. *Journal of General Internal Medicine*, *21*(5), 419–423. https://doi.org/10.1111/J.1525-1497.2006.00420.X/METRICS

Garvin-Doxas, K., & Barker, L. J. (2004). Communication in Computer Science Classrooms: Understanding Defensive Climates as a Means of Creating Supportive Behaviors. *ACM Journal on Educational Resources in Computing*, *4*(1), 2. https://doi.org/10.1145/1060071.1060073

Giibs, A. (1997). Focus Groups. *Social Research Update*. https://sru.soc.surrey.ac.uk/SRU19.html

Ginsberg, M. B., & Wlodkowski, R. J. (2019). Intrinsic Motivation as the Foundation for Culturally Responsive Social-Emotional and Academic Learning in Teacher Education. *Teacher Education Quarterly*, *46*(4), 53–66.

Jr, R. B. F., & Amoloza, E. M. (2015). Addressing Programming Anxiety among Non-Computer Science Distance Learners: A UPOU Case Study. *International Journal for Educational Media and Technology*, *9*(1). https://ijemt.org/index.php/journal/article/view/247

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, *2*(1), 32–37. https://doi.org/10.1145/1929887.1929902

Lin, M. H., Chen, H. C., & Liu, K. S. (2017). A study of the effects of digital learning on learning motivation and learning outcome. *Eurasia Journal of Mathematics, Science and Technology Education*, *13*(7), 3553–3564. https://doi.org/10.12973/EURASIA.2017.00744A

Lindau, M., Almkvist, O., & Mohammed, A. H. (2016). Effects of Stress on Learning and Memory. *Stress: Concepts, Cognition, Emotion, and Behavior: Handbook of Stress*, 153–160. https://doi.org/10.1016/B978-0-12-800951-2.00018-2

Lister, R., Fidge, C., & Teague, D. (2009). Further evidence of a relationship between explaining, tracing and writing skills in introductory programming. *Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE*, 161–165. https://doi.org/10.1145/1562877.1562930

Martin, F., Lee, I., Lytle, N., Sentance, S., & Lao, N. (2020). Extending and evaluating the use-modify-create progression for engaging youth in computational thinking. *SIGCSE 2020 - Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 807–808. https://doi.org/10.1145/3328778.3366971

Perrenet, J., Groote, J. F., & Kaasenbrood, E. (2005). Exploring students' understanding of the concept of algorithm. *ACM SIGCSE Bulletin*, *37*(3), 64–68. https://doi.org/10.1145/1151954.1067467

Piteira, M., & Costa, C. (2013). Learning computer programming: Study of difficulties in learning programming. *ACM International Conference Proceeding Series*, 75–80. https://doi.org/10.1145/2503859.2503871

Plummer-D'Amato, P. (2013a). Focus group methodology Part 1: Considerations for design. *Https://Doi.Org/10.12968/Ijtr.2008.15.2.28189*, *15*(2), 69–73. https://doi.org/10.12968/IJTR.2008.15.2.28189

Plummer-D'Amato, P. (2013b). Focus group methodology Part 1: Considerations for design. *Http://Dx.Doi.Org/10.12968/Ijtr.2008.15.2.28189*, *15*(2), 69–73. https://doi.org/10.12968/IJTR.2008.15.2.28189

Rabiee, F. (2004). Focus-group interview and data analysis. *Proceedings of the Nutrition Society*, *63*(4), 655–660. https://doi.org/10.1079/PNS2004399

Rahardjo, W., Juneman, J., & Setiani, Y. (2013). Computer Anxiety, Academic Stress, and Academic Procrastination on College Students. *Journal of Education and Learning (EduLearn)*, *7*(3), 147–152. https://doi.org/10.11591/EDULEARN.V7I3.179

Rudland, J. R., Golding, C., & Wilkinson, T. J. (2020). The stress paradox: how stress can be good for learning. *Medical Education*, *54*(1), 40–45. https://doi.org/10.1111/MEDU.13830

Saeed, S., & Zyngier, D. (2012). How Motivation Influences Student Engagement: A Qualitative Case Study. *Journal of Education and Learning*, *1*(2), p252. https://doi.org/10.5539/JEL.V1N2P252

Salac, J., Thomas, C., Butler, C., Sanchez, A., & Franklin, D. (2020). Tipp&see: A learning strategy to guide students through use modify scratch activities. *SIGCSE 2020 - Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 79–85. https://doi.org/10.1145/3328778.3366821

Salguero, A., Griswold, W. G., Alvarado, C., & Porter, L. (2021). Understanding Sources of Student Struggle in Early Computer Science Courses. *ICER 2021 - Proceedings of the 17th ACM Conference on International Computing Education Research*, 319–333. https://doi.org/10.1145/3446871.3469755

Schnotz, W., Fries, S., & Horz, H. (2009). Some Motivational Aspects of Cognitive Load Theory. *Contemporary Motivation Research: From Global to Local Perspectives*, 86–113. https://www.researchgate.net/publication/273016902_Some_Motivational_Aspects_of_Cognitive_Load_Theory

Sentance, S. (2021). Teaching programming with PRIMM: the importance of classroom talk. *Teaching Programming with PRIMM: The Importance of Classroom Talk*, *1*. https://www.raspberrypi.org/app/uploads/2022/08/Teaching_Programming_with_PRIMM-1.pdf

Sentance, S., & Waite, J. (2017). PRIMM: Exploring pedagogical approaches for teaching text-based programming in school. *ACM International Conference Proceeding Series*, 113–114. https://doi.org/10.1145/3137065.3137084

Sentance, S., Waite, J., & Kallia, M. (2019a). Teaching computer programming with PRIMM: a sociocultural perspective. *Computer Science Education*, *29*(2–3), 136–176. https://doi.org/10.1080/08993408.2019.1608781

Sentance, S., Waite, J., & Kallia, M. (2019b). Teaching computer programming with PRIMM: a sociocultural perspective. *Computer Science Education*, *29*(2–3), 136–176. https://doi.org/10.1080/08993408.2019.1608781

Shabani, K. (2016). Applications of Vygotsky's sociocultural approach for teachers' professional development. *Cogent Education*, *3*(1). https://doi.org/10.1080/2331186X.2016.1252177

Singh, K., Granville, M., & Dika, S. (2002). Mathematics and Science Achievement: Effects of Motivation, Interest, and Academic Engagement. *The Journal of Educational Research*, *95*(6), 323–332. https://doi.org/10.1080/00220670209596607

Sinta, I., Rahayu, D., & Purnawarman, P. (2019). The Use of Quizizz in Improving Students' Grammar Understanding through Self-Assessment. *Proceedings of the Eleventh Conference on Applied Linguistics (CONAPLIN 2018)*, 102–106. https://doi.org/10.2991/CONAPLIN-18.2019.235

Song, D., Hong, H., & Oh, E. Y. (2021). Applying computational analysis of novice learners' computer programming patterns to reveal self-regulated learning, computational thinking, and learning performance. *Computers in Human Behavior*, *120*, 106746. https://doi.org/10.1016/J.CHB.2021.106746

Von Hausswolff, K., Eckerdal, A., & Thuné, M. (2020). Learning to program hands-on: A controlled study. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3428029.3428058

Wu, H., Li, S., Zheng, J., & Guo, J. (2020). Medical students' motivation and academic performance: the mediating roles of self-efficacy and learning engagement. *Https://Doi.Org/10.1080/10872981.2020.1742964*, *25*(1), 1–8. https://doi.org/10.1080/10872981.2020.1742964

Wu, Z. (2019). Academic Motivation, Engagement, and Achievement among College Students. *College Student Journal*, *53*(1), 99–112.

Zainuddin, Z., & Halili, S. H. (2016). Flipped Classroom Research and Trends from Different Fields of Study. *International Review of Research in Open and Distributed Learning*, *17*(3), 313–340.

Contact email: tevya.tl@gmail.com

Appendices

Appendix A

CURRICULUM SPECIFICATIONS
COMPUTER SCIENCE 2 (SC025)

Topic 1: Introduction to Programming

a) Define programming language, programming paradigm and language translators.
b) Differentiate paradigms of programming language: procedural, object oriented and logic.
c) Differentiate types of language translator: compiler, interpreter and assembler.

Topic 2: Approach in Problem Solving

a) Describe steps in problem solving: problem analysis, design a solution, implementation, testing and documentation.
b) Identify input, process and output based on the given problem statement.

Topic 3: Design a Solution

a) Describe algorithm: pseudocode and flowchart.
b) Explain control structures (sequence, selection: single, dual, multiple, repetition: counter-controlled, sentinel-controlled) using algorithm.
c) Apply appropriate control structures in computational problem solving.

Topic 4: Java Language

4.1 Introduction to Java Program
a) Describe Object-Oriented Programming: class, object, method.
b) Describe the components of a Java program: comments, class, main method, body.
c) Use output and input statements including output string.
d) Construct simple program to perform simple computations.

4.2 Identifiers, data type, operator and expression
a) Identify identifier, variable, constant and reserved word.
b) Identify various primitive data types: int, float, double, boolean and char.
c) Differentiate primitive data types and their usage.
d) Identify various operators: arithmetic, relational and logical.
e) Identify various expressions: arithmetic, relational and logical.
f) Determine the operator precedence and associativity of operators
g) Convert algebraic expression into Java statement.
h) Construct simple programs using primitive data types, operators and expressions.

4.3 Use of control structure
a) Construct program segment by using sequence control structure.
b) Construct program segment by using selection control structure.
c) Construct program segment by using repetition (while, for) counter-controlled or sentinel-controlled structure.
d) Construct program by using sequence control structure.
e) Construct program by using selection control structure.
f) Construct program by using repetition (while, for) counter-controlled or sentinel-controlled structure.
g) Explain types of programming errors: syntax, run-time and logic.

4.4 Array
a) Explain an array and its components.
b) Declare array reference variables and create array.
c) Access array elements – initialize, input, process and output.
d) Construct programs that perform one-dimensional arrays operations for problems involving linear search.
e) Construct programs that perform one-dimensional array operations for problems involving total and frequency.

CURRICULUM SPECIFICATIONS
COMPUTER SCIENCE 2 (SC025)

4.5 Method
4.5.1 Introduction to Java method
a) Explain the meaning and advantages of method.
b) Explain types of method: predefined and user-defined.

4.5.2 Predefined method
a) Identify the use of common predefined method: pow(x,y), sqrt(x).

4.5.3 User-defined method
a) Explain two (2) types of user-defined methods.
b) Explain the general structure of user-defined method.
c) Define and write a method with a return value, and with formal parameters.
d) Define and write a method with a return value, and without formal parameters.
e) Define and write a method without a return value and with formal parameters.
f) Define and write a method without a return value and without formal parameters.
g) Call methods with a return value.
h) Call methods with no return value.
i) Write a method to perform simple computations.

4.6 Java Programs
a) Construct programs that perform one-dimensional array operations for problems involving average, maximum and minimum.
b) Write a program that perform one-dimensional array operations for problems involving average, maximum and minimum.

Appendix B

https://docs.google.com/document/d/1w7M0vtA69dJA4RwPG_D3NHC3GBORpVe_SQfM22UfHdw/edit?usp=sharing

Appendix C

https://docs.google.com/document/d/1yYlUu8BQXxZXAg-Uwrb6mXgeResWWsbL/edit?usp=sharing&ouid=108257376393913038180&rtpof=true&sd=true

Appendix D

Interview Protocol Project	: PRIMM Model towards Malaysian Matriculation Students' Motivation in Learning Programming
Time of Interview	: (To be Confirmed)
Date	: (To be Confirmed)
Place	: Online (Google Meet)
Interviewer	: Researcher - Tevya Letchumanan
Interviewee	: Students
Questions	:

1. Would you be able to describe you experience with the PRIMM module in the computer science classrooms with as much details as possible.
2. You have experienced the traditional classroom approach when learning the first two chapters and the PRIMM approach to learn the other two. Which way do you prefer to learn computer science? – Why?
3. How helpful was PRIMM model to increase your participation in classroom activities?
4. How was the PRIMM module prepared helpful in improving your communication and collaboration amongst your peers?
5. How did the PRIMM module help managing the stress of learning computer science?
6. What are, if any, the challenges of learning computer science in a PRIMM classroom?
7. Overall, in what ways has the PRIMM model that you experienced managed to improve your motivation in learning computer science at matriculation level?

Appendix E

Hyperlink to the transcription of the interview with focus group 1 :
https://drive.google.com/file/d/1pxHR6PRESemWJdpVcyy6h9ELy508Tp29/view?usp=sharing

Hyperlink to the transcription of the interview with focus group 2 :
https://drive.google.com/file/d/1xfsFYDupS9Zv1emg3OqZtipN-kznIxhI/view?usp=sharing

Hyperlink to the transcription of the interview with focus group 3 :
https://drive.google.com/file/d/1lj0euMSHGvGmUSqv4aouAZ7lrEKXAxEm/view?usp=sharing

Hyperlink to the transcription of the interview with focus group 4 :
https://drive.google.com/file/d/1V_v5YahTGynkmmtttdsyR1jgZMwnY7jr/view?usp=sharing

Appendix F

https://docs.google.com/document/d/1jponvvlTQ2H4bp14e86IOo-_-JTqWNCjKzsRIlj7-NM/edit?usp=sharing