Universiti Teknologi MARA
Cawangan Perlis
Jurnal Intelek

Volume 18 Issue 1 (February) 2023

Identification of Student-Related Factors Influencing Programming Courses Learning at UiTM Cawangan Pulau Pinang using Factor Analysis

Naemah Abdul Wahab[1*], Maisurah Shamsuddin[2], Jamal Othman[3], Rozita Kadar[4], Siti Balqis Mahlan[5]

[1,3,4]*Computing Sciences Studies, College of Computing, Informatics and Media, Universiti Teknologi MARA Cawangan Pulau Pinang, Malaysia*
[2,5]*Mathematical Sciences Studies, College of Computing, Informatics and Media, Universiti Teknologi MARA Cawangan Pulau Pinang, Malaysia*

*Authors' Email Address: *[1]naema586@uitm.edu.my, [2]maisurah025@uitm.edu.my, [3]jamalothman@uitm.edu.my, [4]rozita231@uitm.edu.my, [5]sitibalqis026@uitm.edu.my*

Received Date: 22 November 2022
Accepted Date: 11 December 2022
Revised Date: 19 December 2022
Published Date: 31 January 2023

*Corresponding Author

ABSTRACT

Many people claim that teaching and studying computer programming is challenging, mainly for learners who are not from computer science background. It might be difficult for instructors to help students grasp programming principles and acquire the necessary programming abilities to solve problems in the everyday life. Students' lack of logical, creative and critical thinking leads to flaws in problem-based learning implementation (PBL). Based on related past studies, one entity of computer programming learning challenges, namely the student, was examined and analyzed. An online questionnaire was created to collect data from 241 diploma and degree students from the Faculty of Mechanical Engineering (FKM) and the Faculty of Civil Engineering (FKA) who have completed programming courses at Universiti Teknologi MARA (UiTM) Cawangan Pulau Pinang. An analysis using descriptive statistics and factor analysis was applied to determine the student-related component that has the greatest impact on students' learning of programming subjects. The result showed that although some students experience frustration when their lecturer is absent or the conversation is interrupted, the majority of students do not experience any problems with their interest in programming languages or their study habits. Additionally, students' interests and habits were observed as the most influential variables in their decision to learn programming. They are always prepared in class and put a great effort into completing the assessments assigned to them. The findings can be applied to enhance the instructional methodology for programming classes to improve students' comprehension and problem-solving abilities in these subjects.

Keywords: *engineering students, novices, programming difficulty, students' factor*

INTRODUCTION

In higher education, computer programming courses are among the most popular prerequisites for study plans, not just in the domains of engineering, physics, and mathematics, but also of computer science

ISSN: 2231-7716 / E-ISSN 2682-9223
DOI: http://10.24191/ji.v18i1.20457

and information technology. This is due to the fact that it is critical to comprehend computer programming and technology in a number of fields today in order to meet industrial demands. However, learning computer programming is extremely difficult and presents significant obstacles to many students.

According to Moström (2011), the problem must be understood, a solution must be designed using conventional problem-solving techniques, and finally, the answer must be written in a programming language so that a computer can interpret the instructions Furthermore, Abdul Rahman et al. (2018) stated that students need to be able to comprehend the language structure and style, translate requirements into algorithms, find errors, apply the right logic, and learn to use a program development environment in addition to learning the fundamental concepts of programming language syntax.

Therefore, this study is conducted to determine the student-related factors that motivate novice students to learn computer programming. This study aims to increase student interest in programming, enhance programming performance, and help educators of computer science develop better teaching methods for basic programming subjects.

The role of students in learning programming

Deficiency of problem-solving abilities and limited knowledge of program ideas, learning computer programming can be difficult. According to Zhao et al. (2021), learning to program is a difficult cognitive task. A variety of computer languages are needed for programming, and each one contains abstract notions, a deep history, and strict logic. As a result, beginners often encounter misconceptions while learning to program because they typically approach program writing row by row rather than using purposeful structured programs as mentioned by Abdul Rahman et al. (2018). According to a study by Mohamed Shuhidan et al. (2011), beginners think programming is extremely tough since it takes a greater level of expertise than can be attained all at once. As a result, the majority of them start building computer programs without fully analyzing, designing, and understanding the problem, as suggested by Oroma et al. (2012b).

Most students take the easy route by copying their peers' assignment because they lack the necessary problem-solving skills. Worse yet, according to Siti Rosminah & Ahmad Zamzuri (2012), a few of the students merely expect grades from their programming course lecturers out of compassion. Additionally, they found that students' challenges with programming are brought on by three interconnected problems: (1) a challenge in comprehending the basic ideas of programming structure; (2) concerns with task design; and (3) syntax issues with programming languages. They went on to say that beginners lack the skills necessary to master a variety of processes, including planning, analyzing, developing, altering, compiling, and debugging computer codes, in addition to the capacity to employ abstract programming concepts to address particular problems. Besides, Yusoff et al., (2020) stated that there are various types of challenges in learning programming, including cognitive challenges, challenges in designing solutions, developing algorithms, writing and evaluating programs, combining syntax and semantics, challenges related to programming concepts, and limited programming skills. The study evaluated the relationship between these difficulties and computational thinking as well as the significance of computational thinking for overcoming the difficulties associated with learning programming. The study's findings were expected to aid both instructors and students, particularly in the construction of an effective programming teaching and learning approach.

Furthermore, new students are not familiar with the syntax and techniques of programming. Mohamed Shuhidan et al. (2011) identified semicolons and curly bracket syntax errors as the three most frequent beginner errors, along with issues with program design and fundamental program structure. Students also commit syntax errors in their code as stated by Qian & Lehman (2019), including failing to define variables, utilizing incorrect Boolean expressions, and substituting the assignment operator (=) for the comparison operator (==). According to Abdul Rahman et al. (2018), beginners are unable

to solve their program flaws on their own because they are unfamiliar with programming syntax, which prevents them from understanding bugs issued by the compiler. The fundamental grammar, structure, and style of a programming language must be taught to students over time. Programming constructs and abstract data types are challenging for students to comprehend as stated by Mhashi & Alakeel (2013). In addition to the theoretical lectures, they said that before designing computer program, students needed practical lessons to understand how to apply these concepts.

Another element that makes it harder for students to learn about programming is poor learning habits. Each individual learns in a unique way. While some students prefer solo learning, others choose group discussions. The most important aspect, regardless of the sort of learning strategy used, is how students think. Rahmat et al. (2012) claim that learning to program demands a particular attitude. The majority of people find it simpler to learn a subject that they are already familiar with than to grasp a new one, according to a study by Oroma et al. (2012a). Since learning is frequently built on existing knowledge and experiences, many students find computer programming to be challenging because it is unconnected to any previous courses they have studied in elementary or secondary school. According to Cheah (2020), fields in programming demand persistence, ongoing learning, and sporadically applying knowledge from other disciplines. Students typically ask for assistance or give up when they are faced with a challenge. In addition, research by Rahmat et al. (2012) found that many learners show minimal effort in learning to program and instead depend on help from instructors, friends, and the Internet to replicate computer program without fully understanding the work. Even though there aren't enough reference books to help students improve their understanding of computer programming, newcomers exclusively depend on the class materials, slides, and sample answers created by the lecturers.

Finally, Gomes et al. (2012) found in a study that there is a connection between student motivation and grades in programming courses. Due to their unfavorable opinions of programming, the majority of students put out limited effort and lack motivation to learn it with an open mind. This method has resulted in bad marks in basic programming subjects, along with a deficiency in problem-solving abilities. According to Cheah (2020), students have a negative psychological and belief perception of programming courses. The transmission of comments and viewpoints from their seniors who have previously studied the topics is what leads to the implication of unfavorable perceptions towards computer programming disciplines. Students' perceptions of programming as difficult have an effect on both their attitudes toward computer programming and their intrinsic drive. People's behaviors, which refer to the cognitive action that determines whether or not they succeed in finishing an assignment or a course, influence their attitude. Furthermore, Cheah (2020) also added that in computer programming, students' attitudes play a big part in motivating and making a good impression.

Programming subjects demand perseverance and continual learning from time to time; therefore, having a good mindset is essential. As a result, it is critical to ensure that a positive attitude towards computer programming courses is instilled from the beginning of a programming course. Aside from that, constant learning support from lecturers is required to overcome negative perceptions of the difficulties in the subject. As a result, Gomes et al. (2012) recommended that the motivation levels of their students are closely monitored by programming instructors, who make an effort to motivate them by using innovative teaching strategies. This can be achieved by increasing the frequency of practical hands-on sessions and mutual collaborative tutorial discussions on the core ideas of programming, as opposed to delivering the theory through uninteresting lectures. They did stress that this teaching method is best suited for a small group of students in a class of no more than 20 students. By using this method, instructors can learn more about their strengths and weaknesses in computer programming. The summary, which is depicted in Table 1, presents the results of the previous observations.

Table 1: Summary of Problems Encountered by Students in Learning Programming Subjects

Problem	Author(s)	Descriptions
Lack of ability to solve problems and inadequate comprehension of program principles.	(Abdul Rahman et al., 2018; Mohamed Shuhidan et al., 2011; Oroma et al., 2012a; Siti Rosminah & Ahmad Zamzuri, 2012; Yusoff et al., 2020; Zhao et al., 2021)	Since programming involves extensive information and experience to be understood all together, beginners think it is extremely tough. Students' problem-solving skills are lacking, making it difficult for them to perceive the problem correctly. Novices have limited capacity to grasp numerous processes like planning, analysing, developing, altering, compiling, and debugging program codes. They also cannot employ abstract programming concepts to address a specific problem.
Unfamiliar with programming syntax, structure and lack of understanding of debugging errors.	(Abdul Rahman et al., 2018; Mhashi & Alakeel, 2013; Mohamed Shuhidan et al., 2011; Qian & Lehman, 2019)	Beginners frequently make mistakes with basic program structure, including syntax errors, issues with program design, and problems with basic program structure. Since they are unfamiliar with programming syntax, beginners are unable to comprehend the compiler's debug error messages, which prevents them from being able to fix their program errors without assistance from their peers or teachers.
Poor learning style and students' attitudes.	(Cheah, 2020; Oroma et al., 2012a; Rahmat et al., 2012)	The mindset needed to learn to program must be different. Persistence, ongoing learning, and the occasional use of information from other disciplines are all necessary for programming. However, because computer programming has no connection to any of the previous subjects, they took in school, many students believe that this novel subject is challenging. The majority of students show little effort when learning to program and instead depend on help from their instructors, friends and copy the source code from the internet to fulfil programming assignments.
Low grades, negative perception and student motivation.	(Cheah, 2020; Gomes et al., 2012)	The relationship between student motivation and grades in programming courses. Due to their unfavourable opinions of programming, the majority of students put out minimum effort and lack motivation to learn it. Individuals' cognitive actions, which control whether they succeed or fail in finishing an assignment and a course, shape their attitude.

In the following section, the survey conducted is presented with the main objective of obtaining the views of students related to the role of students in influencing programming learning.

METHODOLOGY

A total of 241 students who took programming courses at UiTM Cawangan Pulau Pinang participated in this study. Students from the Faculty of Mechanical Engineering (FKM) and the Faculty of Civil Engineering who were pursuing diplomas and degrees participated in this online survey. After completing the 14-week lectures, students were asked to respond to all questionnaires related to this course. Figure 1 displays the percentage of students who have taken the programming language for the first time and who repeated it. It was discovered that this subject's repeat rate, which only engaged 5.8% of students, wasn't very high.

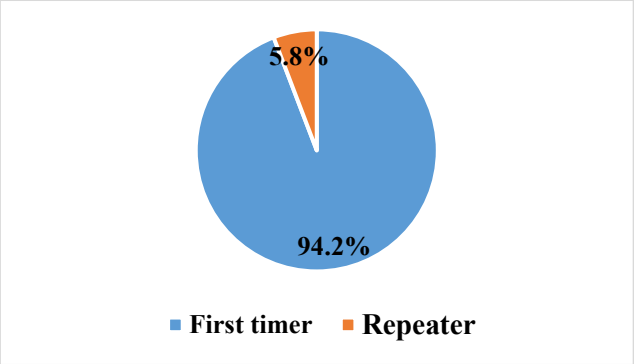

Figure 1: Percentage of First Timer and Repeater Students

The study's questionnaire was divided into two sections, as indicated in Tables 2 and 3. The first section covered the course information, and the second section concentrated on issues relating to the programming course's students. There were 15 items in the second section of the survey. Student-related aspects emphasized more on student interests (5 items) and their habits (10 items).

Table 2: Construct Questions in Course Information

Construct	Options
Program Code	Mechanical Engineering Civil Engineering
Study Level	Diploma Degree
Semester	1/2/3/4/5/6/7/8
Programming Code Taken	CSC128 CSC425
Status Taken	First Timer Not First Timer

Table 3: Construct Questions in Student-Related Factors

Construct	Statements
A - Interest	I make myself prepared for the programming subject.
	I listen attentively to the lecture of my programming lecturer.
	I actively participate in the discussion, answering exercises and/or clarifying things I did not understand.
	I want to get a good grade on tests, quizzes, assignments and projects.
	I get frustrated when the discussion is interrupted, or the lecturer is absent.
B - Study Habits	I do my assignment regularly.
	I put more effort when I do difficult assignments.
	I spend my free time doing assignments or studying.
	I study the lessons I missed if I was absent from the class.
	I study and prepare for quizzes and tests.
	I study harder to improve my performance when I get low grades.
	I spend less time with my friends during school days to concentrate more on my studies.
	I prefer finishing my studying and my assignments first before watching any television program.
	I see to it that extracurricular activities do not delay my studies.
	I have a specific place of study at home that I keep clean and organised.

These surveys employed a five-point Likert scale with responses ranging from 5-always, 4-often, 3-sometimes, 2-rarely, and 1-never. Every statement implies that statements with values higher than 3 are positive and those with values lower than 3 are negative. Prior to performing the questionnaire analysis, the reliability test, also known as Cronbach's Alpha, is conducted. The term "reliability" refers to the accuracy and precision with which a variable is measured by a research instrument. The better the

instrument's reliability, the fewer errors it generates (Kumar, 2018). As seen in Table 4, the value was more than 0.8, which is considered reliable.

Table 4: Reliability Test

Cronbach's Alpha	N of items
.830	15

The collected data were analyzed using descriptive statistics and factor analysis (FA). Analysis using mean values and standard deviations was used to generally identify student-related factors that influence programming learning. Next, FA was conducted to find the most important factors influencing programming learning. Kaiser-Meyer Olkin (KMO) statistics were measured first before FA was performed. The KMO value that should be used is at least 0.5. After that, a Bartlett sphere test should be performed, and the result must be less than 0.05 to proceed with FA. At eigenvalues greater than one, the optimal number of components will be selected. Finally, the factor varimax rotation approach was used to generate the best factor.

RESULTS AND DISCUSSION

Table 5 displays the findings for the mean and standard deviation of student factors associated with learning a programming language. Except for item A5, which had the lowest mean, it was discovered that all items had a mean greater than 3. (3). For item A5, the standard deviation score (1.076) was likewise fairly high, indicating that students' replies were not reliable. Most pupils expressed interest in learning a programming language. They pay close attention to the lecture because they desire to do well in this topic. Based on a minimal standard deviation, the answer ratings provided were likewise quite consistent.

Table 5: Student-Related Factors

Item no.	Statement	Mean	SD
A1	I make myself prepared for the programming subject.	3.35	0.897
A2	I listen attentively to the lecture of my programming lecturer.	4.22	0.656
A3	I actively participate in the discussion, answering exercises and/or clarifying things I did not understand.	3.54	0.831
A4	I want to get a good grade on tests, quizzes, assignments and projects.	4.73	0.598
A5	I get frustrated when the discussion is interrupted, or the lecturer is absent.	3.00	1.076
B1	I do my assignment regularly.	3.96	0.836
B2	I put more effort when I do difficult assignments.	4.23	0.813
B3	I spend my free time doing assignments or studying.	3.6	0.827
B4	I study the lessons I missed if I was absent from the class.	4.07	0.779
B5	I study and prepare for quizzes and tests.	4.39	0.693
B6	I study harder to improve my performance when I get low grades.	4.37	0.639
B7	I spend less time with my friends during school days to concentrate more on my studies.	3.36	0.869
B8	I prefer finishing my studying and my assignments first before watching any television program.	3.73	0.929
B9	I see to it that extracurricular activities do not delay my studies.	3.44	0.912
B10	I have a specific place of study at home that I keep clean and organised.	4.05	0.988

Considering the items for student habits (B1 to B10), the highest mean score was on Item B5 (4.39) and the lowest mean score was on Item B7 (3.36). Overall, the majority of students showed the habit of learning programming languages very well. They often study and prepare for tests and quizzes (mean = 4.39), study harder to improve their performance when getting low grades (mean =4.37) and work harder when doing difficult tasks (mean = 4.23). They also sometimes spend less time with friends during semester breaks and participate in co-curricular activities to ensure that all these activities do not interfere with their studies. In conclusion, student-related factors greatly influence programming learning as the majority of students showed a positive response to their interest and habits of learning programming, which means that they are always ready and study hard to complete a given assessment.

Table 6. KMO and Bartlett's Test

KMO and Bartlett's Test		
Kaiser-Meyer-Olkin Measure of Sampling Adequacy.		.850
Bartlett's Test of Sphericity	Approx. Chi-Square	958.005
	df	105
	Sig.	.000

FA was then used to find the main factor that explains the dependent variables that most influence programming learning. The result of KMO and Bartlett's Test is shown in Table 6. The obtained value of KMO was 0.850 (> 0.5), which is higher than 0.5. Bartlett's test yielded a p-value of 0.000 (<0.05), which is less than a specified value of 0.05. As a result, there was a relationship between the variables. This implies that the data is suitable for FA.

Figure 2 shows the 15 components that have been identified and the eigenvalues obtained after extraction. It was found that 3 components were considered as factors based on eigenvalues greater than 1.

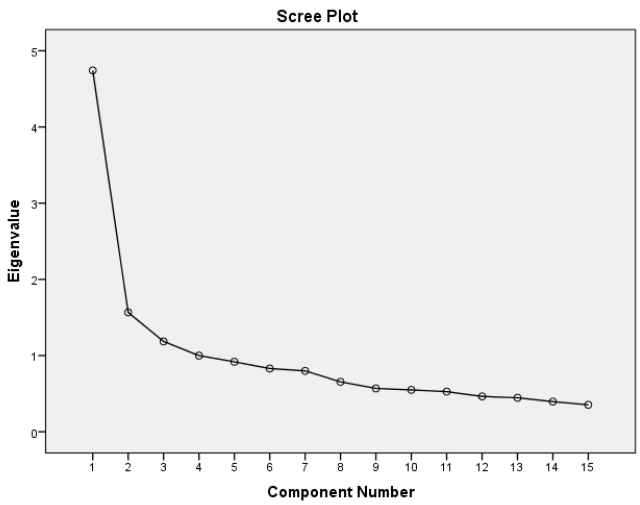

Figure 2: Relationship between Component Number and Eigenvalue

Table 7 then displays the total variation explained for these three components. The total cumulative variances were discovered to be 49.963% that is, the first component explained a total of 30.616%, the second factor explained 10.444%, and the third factor explained 7.904%.

Table 7. Total Variance Explained

Total Variance Explained									
Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings		
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	4.742	31.616	31.616	4.742	31.616	31.616	2.791	18.609	18.609
2	1.567	10.444	42.059	1.567	10.444	42.059	2.560	17.063	35.673
3	1.186	7.904	49.963	1.186	7.904	49.963	2.144	14.290	49.963

Similar to the extraction approach, the eigenvalues in the rotational stage must be greater than 1. The first factor's variance decreased by 13.007%, from 31.616% to 18.609%. The variances of the other two variables, however, went up by around 25.229% and 42.059%. These proved that, in comparison to other factors, the first component was more correlated with the dependent variable.

Based on the results in Table 8, the main factors or component 1 that influence learning programming, which contained items A1, A2, A3, B1, B2 and B3. This was followed by the second factor, which contained items A4, B4, B5, B6 and B10. The last factors were from items A5, B7, B8 and B9.

Table 8. Rotated Component Matrix

Component	1	2	3
A1	0.576		
A2	0.688		
A3	0.733		
B1	0.555		
B2	0.596		
B3	0.555		
A4		0.563	
B4		0.572	
B5		0.735	
B6		0.765	
B10		0.53	
A5			0.694
B7			0.61
B8			0.567
B9			0.715

Overall, the results showed that the contributing factors for students to learn programming are A1 (preparing for programming subjects), A2 (listening attentively to lectures), A3 (participating actively in discussions, answering exercises, and clarifying things that are not understood), B1 (does tasks often), B2 (puts more effort when doing difficult tasks) and B3 (spends free time doing tasks or studying).

CONCLUSION

In conclusion, although the majority of students showed a positive response to their interest and habits of learning programming, novice students are lacking in computer programming skills. Creating creative teaching methods would enhance students' programming abilities. To improve students' programming skills, instructors should switch from traditional face-to-face instruction to collaborative online instruction, which is more engaging and effective.

The problem-based learning (PBL) strategy is one of the most effective ways to improve problem-solving abilities. Exposing students to real-world problems and providing solutions using relevant theories could result in increased student competencies, a greater quality of future programmers and the fulfilment of industry demands. Future studies should focus on the best techniques used by educators to improve students' programming abilities so they may learn more about programming utilizing a variety of engaging teaching strategies, tools, platforms, and creativity.

ACKNOWLEDGEMENTS

The authors would like to thank all the students of Universiti Teknologi MARA Cawangan Pulau Pinang who participated in this study.

FUNDING

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

AUTHORS' CONTRIBUTION

Kadar, R., Wahab, N.A. and Othman, J. conceived and planned the experiments. Kadar, R. carried out the experiments and data preparation. Shamsuddin, M. and Mahlan, S.B. contributed to the interpretation of the results. Wahab, N.A. took the lead in writing the manuscript. All authors provided critical feedback and helped shape the research, analysis and manuscript.

CONFLICT OF INTEREST DECLARATION

We certify that the article is the Authors' and Co-Authors' original work. The article has not received prior publication and is not under consideration for publication elsewhere. This research has not been submitted for publication nor has it been published in whole or in part elsewhere. We testify to the fact that all Authors have contributed significantly to the work, validity and legitimacy of the data and its interpretation for submission to Jurnal Intelek.

REFERENCES

Abdul Rahman, T. F. B., Anuar, N., & Mohd Said, R. F. (2018). How the nature of programming and learning materials affects novice learner's motivation and programming ability. *ACM International Conference Proceeding Series*, 124–128. https://doi.org/10.1145/3178158.3178184

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

Gomes, A. J., Santos, Á. N., & Mendes, A. J. (2012). A study on students' behaviours and attitudes towards learning to program. *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*, 132–137. https://doi.org/10.1145/2325296.2325331

Kumar, R. (2018). *Research methodology: A step-by-step guide for beginners*. Sage.

Mhashi, M. M., & Alakeel, A. L. I. M. (2013). Difficulties Facing Students in Learning Computer Programming Skills at Tabuk University. *Recent Advances in Modern Educational Technologies*, 15–24.

Mohamed Shuhidan, S., Hamilton, M., & D'Souza, D. (2011). Understanding novice programmer difficulties via guided learning. *ITiCSE'11 - Proceedings of the 16th Annual Conference on Innovation and Technology in Computer Science*, 213–217. https://doi.org/10.1145/1999747.1999808

Moström, J. E. (2011). *A study of Student Problems in Learning to Program*. http://urn.kb.se/resolve?urn=urn:nbn:se:umu:diva-48216%5Cnhttp://umu.diva portal.org/smash/get/diva2:447104/FULLTEXT02%5Cnhttp://umu.diva-portal.org/smash/record.jsf?pid=diva2:447104

Oroma, J., Wanga, H., & Ngumbuke, F. (2012a). Challenges of Teaching and Learning Computer Programming in a Developing Country: Lessons From Tanzania. *INTED2012 Proceedings*, *October*, 3820–3826. https://doi.org/10.13140/2.1.3836.6407

Oroma, J., Wanga, H., & Ngumbuke, F. (2012b). Challenges of teaching and learning computer programming in developing countries: lessons from tumaini university. *DOI: Https://Doi. Org/10.13140/2.1*, *3836*.

Qian, Y., & Lehman, J. D. (2019). Using Targeted Feedback to Address Common Student Misconceptions in Introductory Programming: A Data-Driven Approach. *SAGE Open*, *9*(4). https://doi.org/10.1177/2158244019885136

Rahmat, M., Shahrani, S., Latih, R., Yatim, N. F. M., Zainal, N. F. A., & Rahman, R. A. (2012). Major Problems in Basic Programming that Influence Student Performance. *Procedia - Social and Behavioral Sciences*, *59*, 287–296. https://doi.org/10.1016/j.sbspro.2012.09.277

Siti Rosminah, M. D., & Ahmad Zamzuri, M. A. (2012). Difficulties in learning Programming: Views of students. *1st International Conference on Current Issues in Education (ICCIE2012)*, *October 2014*, 74–78. https://doi.org/10.13140/2.1.1055.7441

Yusoff, K. M., Ashaari, N. S., Wook, T. S. M. T., & Ali, N. M. (2020). Analysis on the requirements of computational thinking skills to overcome the difficulties in learning programming. *International Journal of Advanced Computer Science and Applications*, *11*(3), 244–253. https://doi.org/10.14569/ijacsa.2020.0110329

Zhao, H., Li, M., Lin, T., Wang, R., & Wu, Z. (2021). ProLog2vec: Detecting novices' difficulty in programming using deep learning. *IEEE Access*, *9*, 53243–53254.