Effective Strategies for Enhancing Student Learning in Programming Education among PBT Students

Norakmar Binti Mohd Nadzari
Jabatan Teknologi Maklumat dan Komunikasi
Politeknik Besut, Terengganu
norakmar@polibesut.edu.my

Nur Azhani Binti Rosly
Jabatan Teknologi Maklumat dan Komunikasi
Politeknik Besut, Terengganu
nurazhani@polibesut.edu.my

Eni Aryanti Binti Yusoff
Jabatan Teknologi Maklumat dan Komunikasi
Politeknik Besut, Terengganu
eni@polibesut.edu.my

Abstract

This research focuses on investigating effective strategies for enhancing student learning in programming education. With the growing demand for programming skills in today's digital era, it is crucial to optimize teaching and learning practices to ensure students develop a solid foundation in programming concepts and skills. The study examines various dimensions of programming education, including instructional methods, learning environments, tools, and techniques, with the goal of identifying approaches that positively impact student engagement, comprehension, and retention.The research explores the effectiveness of different instructional methods, such as traditional lectures, interactive discussions, hands-on coding exercises, and project-based learning, in teaching programming to students at different educational levels. It also investigates the role of the learning environment, including physical classrooms, online platforms, and interactive coding environments, in promoting active learning and student engagement. The findings provide valuable insights and practical recommendations to optimize teaching practices and equip students with the necessary programming skills in today's digital world.

Keywords: *programming, traditional lectures, interactive discussions, hands-on exercises, students' engagement.*

1. Introduction

The investigation of efficient teaching methods for students studying programming is the main goal of this study. To guarantee that students acquire a strong foundation in programming concepts and skills, it is critical to optimize teaching and learning practices considering the rising demand for programming skills in the current digital world. The objective of the study is to find strategies that have a beneficial influence on student engagement, comprehension, and retention. It studies different aspects of programming education, including instructional methods, learning settings, tools, and techniques.

The study examines how well various teaching strategies, including standard lectures, interactive discussions, practical coding activities, and project-based learning, teach programming to students at various educational levels. It also investigates how the learning environment, which includes actual classrooms, digital platforms, and interactive coding environments, may encourage student engagement and active learning. The results offer insightful analysis and useful suggestions for improving instructional strategies and giving students the programming abilities, they need for the modern digital environment. Collaborative learning and peer interaction are also addressed, with a focus on strategies such as collaborative coding projects, pair programming, group discussions, and online forums that foster collaboration, knowledge sharing, and problem-solving skills among programming students (Qureshi et al., 2021). The research further explores effective assessment methods for evaluating student progress and providing timely feedback in programming courses.

To improve student learning experiences in programming education, this research intends to offer useful insights and actionable recommendations for instructors, curriculum makers, and institutions. Educators may give students the skills they need to thrive in the digital era and contribute to the always developing area of programming by using successful tactics.

2. Problem Statements

Finding and putting into practice efficient solutions that improve student learning outcomes is necessary in programming education. Despite the rising demand for programmers, students frequently struggle to grasp difficult programming ideas and master practical coding skills. Even though they are frequently employed, traditional lectures may not always properly engage students or foster deeper comprehension. Similar to this, having few opportunities for interactive conversations and hands-on coding exercises might impede critical thinking and active learning.

The efficiency of various teaching strategies and their effects on student engagement, comprehension, and retention in programming education are also not well studied (Kilag et al., 2023). Without a clear grasp of which strategies work best, instructors may find it difficult to improve their instructional strategies and give students engaging learning opportunities.
Furthermore, in order to address individual gaps and problems, programming education students' different learning demands necessitate personalized approaches. The use of intelligent tutoring systems, personalized education, and adaptive learning technologies to meet the needs of different students and improve their programming learning experiences has, however, received very little attention.

As a result, it is crucial to address the issue and provide practical solutions to improve student learning in programming education. Researchers can enhance teaching strategies, encourage active student engagement, and advance successful learning outcomes in programming education by evaluating instructional approaches, examining novel approaches, and examining the impact of personalized learning strategies.

3. Objective

This study aims to explore effective strategies for enhancing students learning in programming education. In particular, the objectives of the study are as follows:

1. To assess the impact of different instructional methods on student learning outcomes in programming education
2. To identify effective strategies for promoting active learning and student engagement in programming education:
3. To explore the role of personalized learning approaches and adaptive technologies in programming education

4. Methodology

4.1 Sample and data collection method

The method of obtaining data for research conducted is quantitative using questionnaire method. The study of population was 50 students from Diploma Information Technology from semester 1 until 5. Likert scale 4 score is used to measure the level of approval and disapproval of respondents to the statements given.

4.2 Sample and data collection method

The following table indicates the measures of the study variable used in the study. The instruments items consisted of 4 parts which are Demographic, Instructional Method on teaching and learning, students' engagement, and collaborative learning in programming education.

Table 1. Items in questionnaire

Item/Questionnaire	**Description**
Part A: Demographic	
1-3	Respondents' general information
4	Programming Experience
Part B: Instructional Method on Teaching and Learning	
1-9	Different instructional methods on student learning outcomes in programming
Part C: Active Learning and Student Engagement in Programming Education	
1-9	Effective strategies in active learning and student engagement
Part D: Collaborative Learning and Peer Interaction in Programming Education	
1-9	Learning approaches in programming education

5. Finding & Discussion

Out from 50 respondents randomly from DDT students who answered the questionnaires, 68% (n= 34) are male and 32%(n=16) are females. While the majority of respondents from semester 3 is 44%(n=22), followed by 28%(n=11) semester 5, 22%(n=11) from semester 4, 4%(n=2) from semester 2 and 2%(n=1) from semester 1 student.

Table 2 summarizes the level of student's programming proficiency with data presented in terms of frequency and percentage. Only two respondents (2%) lack any programming experience. A total of 21 people (43%) is categorized as beginner programmers, while 26 respondents fit the intermediate category, indicating a moderate level of programming expertise. Only 1 (2% of respondents) are regarded as programming experts. According to these results, the majority of respondents are either beginning or intermediate level.

Table 2: Level of Programming Proficiency

Item	**Frequency**	**Percent (%)**
No programming Knowledge	2	4
Beginner	21	42
Intermediate	26	52
Expert	1	2
Total	50	100

Findings on Objective 1.

The first objective of the study was to evaluate the impact of different teaching methods on student learning outcomes in programming education. The researcher asked 9 questions to the respondents about the different teaching methods in programming which involved classroom teaching, discussion, and practical training. The following is the data obtained from the respondents through the instruments carried out.

Based on the findings obtained, it can be concluded that the respondents agreed teaching and learning in class help them to improve their understanding in the field of programming. In addition, interactive discussion encourages them to ask questions and get the explanation about the concept of programming language. In the meantime, the practical exercise helps them to improve their skills from theoretical concepts to the real-world programming scenario.

The findings on table 3 shows that both means scores for traditional learning and interactive discussions are 3.30, indicate that respondents rated these two method teaching similarly in terms of effectiveness. Respondents agreed that traditional lectures and interactive coding facilitated understanding of complex programming topics. For hands-on coding exercises received a slightly higher mean score of 3.44 found that it could be more effective in conveying programming concepts and principles.

Table 3. Instructional Method on Teaching and Learning

Item	Mean	Standard Deviation
Traditional lectures	3.30	0.666
Interactive discussion	3.30	0.706
Hands-on coding exercises	3.44	0.767

Findings on Objective 2.

The next objective of the study was to identify effective strategies for promoting active learning and student engagement in programming education. The researcher asked 9 questions to the respondents about the student's engagement in programming education. The following is the data obtained from the respondents through the instruments carried out.

Table 4 shows results of mean and SD for effective strategies in active learning and engagement in with the following results, physical classroom environment (3.36), online programming platform (2.47) and interactive coding environments (3.31). Based on the findings obtained on table 4, it can be concluded that the respondents agreed the physical classroom environment influences their motivation in learning programming.

Table 4. Active Learning and Student Engagement in Programming Education

Item	Mean	Standard Deviation
Physical classroom environment	3.36	0.767
Online programming	2.47	0.806
Interactive coding	3.31	0.704

Findings on Objective 3.

The findings on table 5 were related to collaborative and peer interaction in learning programming. Group discussion received the highest mean score of 3.36, indicate that respondents rated group discussion enhance understanding of programming concepts. For collaborative coding project had mean score 3.35, suggesting that respondents also agreed it could facilitate knowledge sharing and learning about programming. Meanwhile for pair programming (3.33) facilitates mutual learning and communication between partners.

Table 5. Collaborative Learning and Peer Interaction in Programming Education

Item	Mean	Standard Deviation
Collaborative coding projects	3.35	0.765
Pair programming	3.33	0.761
Group Discussion	3.36	0.750

6. Conclusion and Future Recommendation

Table 6 provides a list of strategies that have been implemented and rated by students who have taken one or more programming courses to improve student learning in programming education.

Table 6. Teaching Methods and Strategies.

Methods	**Strategies**
Lectures	Explicit Teaching
Laboratories practice	Problem Solving
Discussion	Pair/Group programming

Since one of the hardest and most challenging subjects in the computer science curriculum is computer programming, it's critical to pick a teaching strategy that will provide students with the optimal learning environment.

The teaching and learning of introductory programming courses is a great challenge for all those who have to deal with this area. Programming is a process of transforming a mental plan of current terms into terms compatible with the computer [17]. When teaching computer programming, the main objective is to empower students with the skills needed to create computer programs that can solve real-world problems. (Figueiredo & García-Peñalvo, 2019). It was discovered that the author had modified this approach as a result of the lengthy years (about 20 years) spent instructing programming. In order to boost performance, needs to concentrated on first-year programming students and modified this approach. Concepts are first discussed, followed by the presentation of sample programs and other relevant tasks. After the topics are covered, the students work hard to complete lab work, which is laboratory practice that assesses the students' comprehension of the lessons. This assignment, along with others, demonstrates the students' various approaches to problem-solving and establishes their comprehension of the stages needed in programming.

To further establish that laboratory practice is an essential part of effective teaching strategies. For learners to acquire hands-on skills in programming, teachers are requested to combine laboratory practice sessions, projects, seminar and tutorials along with lectures in order to encourage students to have several opportunities and to adapt to the best practice. This is in agreement to the findings of [Miliszewska & Tan, 2007] which states that; teachers' task is to select an appropriate teaching method or blend of methods.

On the other hand, to give students better possibilities to connect with their classmates and teachers, more than one teaching technique should be adapted while teaching programming. We advise teachers to do this by implementing peer tutoring, pair/group programming, and problem-solving lessons. The majority of participants think that programming is highly difficult since it requires adapting one of these techniques or one of these strategies.

References

Al-Ghazali, M. (2004). Khuluk Al Muslim. Damascus: Dar Alkalam.

Blacking, J. 1973. Music in society and culture. In How musical is man?, 32–53. Washington DC:University of Washington Press. Carey, I. 1976. The Senoi. In Orang Asli the aboriginal tribes Peninsular Malaysia, 129– 213. Kuala Lumpur: Oxford University Press

Clancy, M., Titteron, N., Ryan, C., Slotta, J., & Linn, M., "New roles for students, instructors, and computers in a lab-based introductory programming course." *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, pp. 132-136, 2003.

Figueiredo, J., & García-Peñalvo, F. J. (2019). Teaching and learning strategies of programming for university courses. Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality. https://doi.org/10.1145/3362789.3362926

Hu Xiaohui, "Improving teaching in Computer Programming by adopting student-centred learning strategies", *The China Papers*, pp. 47 – 48, November 2006.

Iwona Miliszewska and Grace Tan. (2007). "Befriending Computer Programming: A Proposed Approach to Teaching Introductory Programming.", Issues in Informing Science and Information Technology Volume 4, pp. 277 – 287

Kilag, K. T., Dejino, J. A., Almendras, R. C., Arcillo, M. T., Mansueto, D. P., & Frances, C. (2023). Enhancing Student Engagement in Multimedia-Mediated Constructivist Learning: Exploring Students' Perceptions. EUROPEAN JOURNAL of INNOVATION in NONFORMAL EDUCATION, 3(6), 51–59. http://inovatus.es/index.php/ejine/article/view/1791

Kirsti Ala-Mutka, "Problems in learning and teaching programming," a literature study for developing visualizations in the Codewitz-Minerva project,pp. 1 – 13, (n.d)

M. Butler and M. Morgan, "Learning challenges faced by novice programming students studying high level and low feedback concepts," Proceedings ascilite Singapore, pp. 99 – 107, 2007.

Miliszewska, I., & Tan, G. (2007). Befriending Computer Programming: A Proposed Approach to Teaching Introductory Programming. Issues in Informing Science and Information Technology, 4, 277–289. https://doi.org/10.28945/950

Qureshi, M. A., Khaskheli, A., Qureshi, J. A., Raza, S. A., & Yousufi, S. Q. (2021). Factors affecting students' learning performance through collaborative learning and engagement. Interactive Learning Environments, 31(4), 1–21. https://doi.org/10.1080/10494820.2021.1884886

Schneider, R. M., Krajcik, J., Marx, R. W., & Soloway, E. (2002). Performance of students in project-based science classrooms on a national measure of science achievement. Journal of Research in Science Teaching, 39(5), 410-422.

S. Mohorovicic and V. Strcic, "An Overview of Computer Programming Teaching Methods," pp. 1 – 6, (n.d).

T. Jenkins, "On the Difficulty of Learning to Program," Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences, Lough- borough, UK, pp. 53-58, 2002.