Uluslararası İleri Doğa Bilimleri ve Mühendislik Araştırmaları Dergisi
Sayı 9, S. 261-271, 2, 2025

Araştırma Makalesi

IJANSER
https://as-proceeding.com/index.php/ijanser
ISSN:2980-0811

International Journal of Advanced Natural Sciences and Engineering Researches
Volume 9, pp. 261-267, 2, 2025

Research Article

Tracking Students' Progress in Introductory C Programming Courses through Moodle Tests with Randomized Questions

Ladislav Végh [1*], Ondrej Takáč [2] and Krisztina Czakóová [3]

[1]*Department of Informatics, Faculty of Economics and Informatics, J. Selye University, Slovakia*
https://orcid.org/0000-0001-9242-4983

[2]*Department of Informatics, Faculty of Economics and Informatics, J. Selye University, Slovakia*
https://orcid.org/0000-0001-8836-0227

[3]*Department of Informatics, Faculty of Economics and Informatics, J. Selye University, Slovakia*
https://orcid.org/0009-0000-3096-9427

[*]*(veghl@ujs.sk) Email of the corresponding author*

(Received: 10 February 2025, Accepted: 17 February 2025)

(2nd International Conference on Recent and Innovative Results in Engineering and Technology ICRIRET, February 11-12, 2025)

ATIF/REFERENCE: Végh, L., Takáč, O. & Czakóová, K. (2025). Tracking Students' Progress in Introductory C Programming Courses through Moodle Tests with Randomized Questions. *International Journal of Advanced Natural Sciences and Engineering Researches*, 9(2), 261-267.

Abstract – Assessing students' progress in introductory programming courses is crucial for identifying learning gaps and improving teaching methods. This study evaluates the effectiveness of Moodle-based tests with randomized questions in monitoring student progress in C programming courses at J. Selye University during the 2023/24 academic year. A series of ten tests were administered across two courses, covering essential programming topics such as data types, variables, conditional statements, loops, two- and three-dimensional arrays, recursion, and sorting algorithms. The results revealed significant variations in student performance, with recursion and the pretest/posttest loops presenting the greatest challenges. The correlation analysis of test scores showed strong relationships among related topics, confirming the structured progression of the curriculum. These findings suggest that Moodle-based assessments offer valuable insights into students' learning trajectories, enabling educators to adapt their instructional strategies accordingly. Such insights can help optimize introductory programming curricula, enhancing student engagement and understanding.

Keywords – C Programming, Introductory Programming, Moodle Tests, Randomized Questions, Student Assessment.

I. INTRODUCTION

Programming and algorithmization are key components of the informatics curriculum for computer science students at J. Selye University. In their first year, undergraduate computer science students learn programming using the C programming language. Given that students come from various high schools, their prior programming experience varies significantly. While some students have solid programming skills, many have no programming experience. As a result, the introductory programming course may be tedious for those with experience, while it presents significant challenges for those with no background in programming and algorithmization. To address this issue, during the 2023/24 academic year, we placed experienced students in a separate group and assigned them more challenging tasks in their first two

introductory programming courses at the university. Due to the other students requiring more assistance with programming, we implemented Moodle tests featuring randomized questions after each main topic of the course to track their progress in greater detail. In this paper, we summarized the results of these Moodle tests and shared our experiences.

II. ENHANCING TEACHING AND LEARNING WITH TECHNOLOGY

For more than 15 years, we have used electronic resources and tools to support the teaching and learning of programming and algorithmization, especially in introductory computer programming courses. We have created various interactive algorithm visualizations and provided electronic learning materials to assist students in their education, helping them acquire programming knowledge and experience [1], [2], [3], [4]. Electronic teaching materials and assignments through e-learning portals have become increasingly important, especially since the COVID-19 pandemic [5]. Learning Management Systems (LMS), such as Moodle, allow educators to share electronic materials and assess students' knowledge effectively. It provides several forms of online assessments, such as quizzes, assignments, forums, and workshops [6]; it can be utilized across a diverse range of subjects. For example, Etemadfar et al. [7] described in their paper how Moodle can be used for teaching and learning intermediate English language. Reina et al. [8] developed Moodle-based open educational resources called PLATA, which is an online platform for chemistry undergraduate fully automated assignments. Huerta-Gomez-Merodio and M. V. Requena-Garcia-Cruz [9] combined the Moodle platform with the FastTest plugin to develop computer coding problems for instructional and evaluative purposes in two engineering courses. Karkina et al. [10] created an online course in Moodle for future music teachers. Their results showed the effectiveness of this course in significantly improving students' professional skills. Popovic et al. [11] studied the effects of integrating Moodle-based blended learning into physiology education. The results showed that this method improved student engagement, attendance, and performance. Hickman and Bell [12] used Moodle with the CodeRunner plugin to automate the assessment of a new high school computer programming standard in New Zealand. Their findings revealed that teachers were enthusiastic about using automated assessments for this standard and were pleased with how the system facilitated the grading process. Kaya and Özel [13] integrated an online compiler and a plagiarism detection tool into Moodle to streamline the process of programming assignments. This integration reduced grading time and discouraged plagiarism. Their study found that this approach significantly decreased plagiarism rates, improved student performance, and saved instructors a substantial amount of time in grading. Gamage et al. [14] evaluated multiple studies regarding Moodle and found that it is primarily utilized within university STEM fields and significantly enhances student performance, satisfaction, and engagement. Moodle is being increasingly adopted as a platform for both adaptive and collaborative learning, as well as for improving online assessments.

III. MATERIALS AND METHOD

To track undergraduate first-year computer science students' performance during the introductory programming courses, we developed several tests in Moodle with randomized questions. We created five tests in the first course (PR1) taught in the winter semester of the academic year 2023/24. The topics of these tests corresponded to the syllabus of the course; they were: (1) data types, variables, conditional statements, (2) iterating with counting loop, (3) using pretest and posttest loops, (4) using array data structures, and (5) operations on arrays. We created ten questions for each test and randomly selected five questions for each student. As an example, one of these questions is shown in Fig. 1.

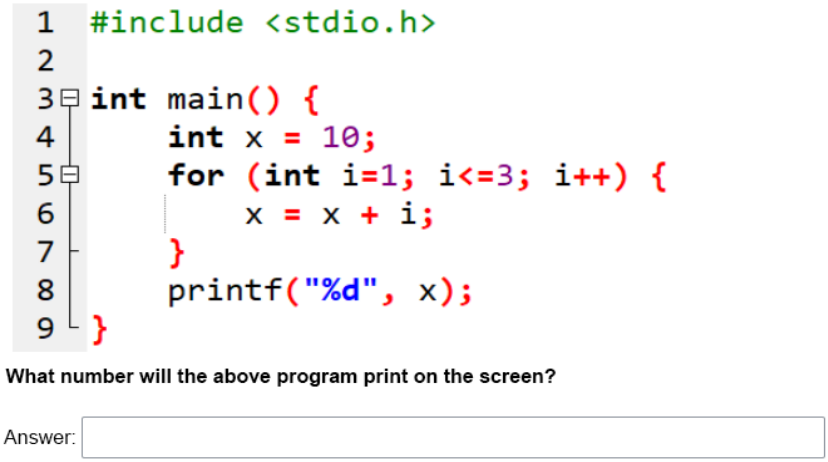

Fig. 1 A question about iterating with counting for loop

During the summer semester of the academic year 2023/24, we developed five tests with ten questions each in the subsequent computer programming course (PR2). Students were given five random questions in each test for their assignments. The topics of these tests were the following: (1) using sorting algorithms with time complexity $O(n^2)$, (2) developing simple recursive functions, (3) finding elements in vectors or matrices, (4) simple operations on matrices, and (5) complex operations on matrices.

Students took tests after completing each topic, typically every two weeks during the courses. After submitting all assignments, we compared the students' average scores on each test. Additionally, we calculated correlation matrices for each computer programming course to identify which topics were most closely correlated.

IV. RESULTS

During the winter semester of the 2023/24 academic year, 84 first-year undergraduate students enrolled in the introductory computer programming course (PR1). After evaluating their prior programming experience, 17 students were placed in a separate group designated for advanced learners. These advanced students were assigned more challenging programming tasks instead of taking the tests outlined in this paper. The remaining 67 students were divided into three groups and completed the tests after each topic. Table 1 summarizes the average scores from these tests.

Table 1. Average test scores for the PR1 course

Test	N	Average
Test #1: Data types, variables, conditional statements	61	70.2%
Test #2: Iterating with for loop	61	55.4%
Test #3: Pretest (while) and posttest (do..while) loops	63	41.0%
Test #4: Array data structure	58	63.8%
Test #5: Operations on arrays	64	65.9%

In the summer semester of the 2023/24 academic year, 82 students were enrolled in a subsequent computer programming course (PR2). Based on their scores in the PR1 course, 23 students were placed in an advanced learners group, where they tackled challenging programming tasks instead of taking tests. The remaining 59 students were divided into three groups and took tests after every topic of the course. The average scores from these tests are presented in Table 2.

Table 2. Average test scores for the PR2 course

Test	N	Average
Test #6: Sorting algorithms with time complexity O(n^2)	52	51.5%
Test #7: Simple recursive functions	41	38.0%
Test #8: Finding elements in vectors or matrices	40	51.0%
Test #9: Simple operations on matrices	46	58.3%
Test #10: Operations on matrices	56	60.4%

The data in the tables show that the most challenging topic for students was recursion, specifically in developing simple recursive functions during the PR2 course, where they achieved an average score of only 38%. The second most difficult topic was using pretest and posttest loops in the PR1 course, with an average score of 41%.

To better understand the results, Fig. 2 and Fig. 3 illustrate the distribution of students' scores for each test. As shown in Fig. 2, the most challenging assessment during the PR1 course was test #3, which focused on pretest and posttest loops. Most students scored either 0%, 20%, or 40% on this test. In contrast, the most manageable assessments were test #1, which covered data types, variables, and conditional statements, as well as test #5, which involved operations on arrays. On test #1, most students scored either 80% or 100%, while on test #5, 27 out of 64 students achieved a score of 100%.

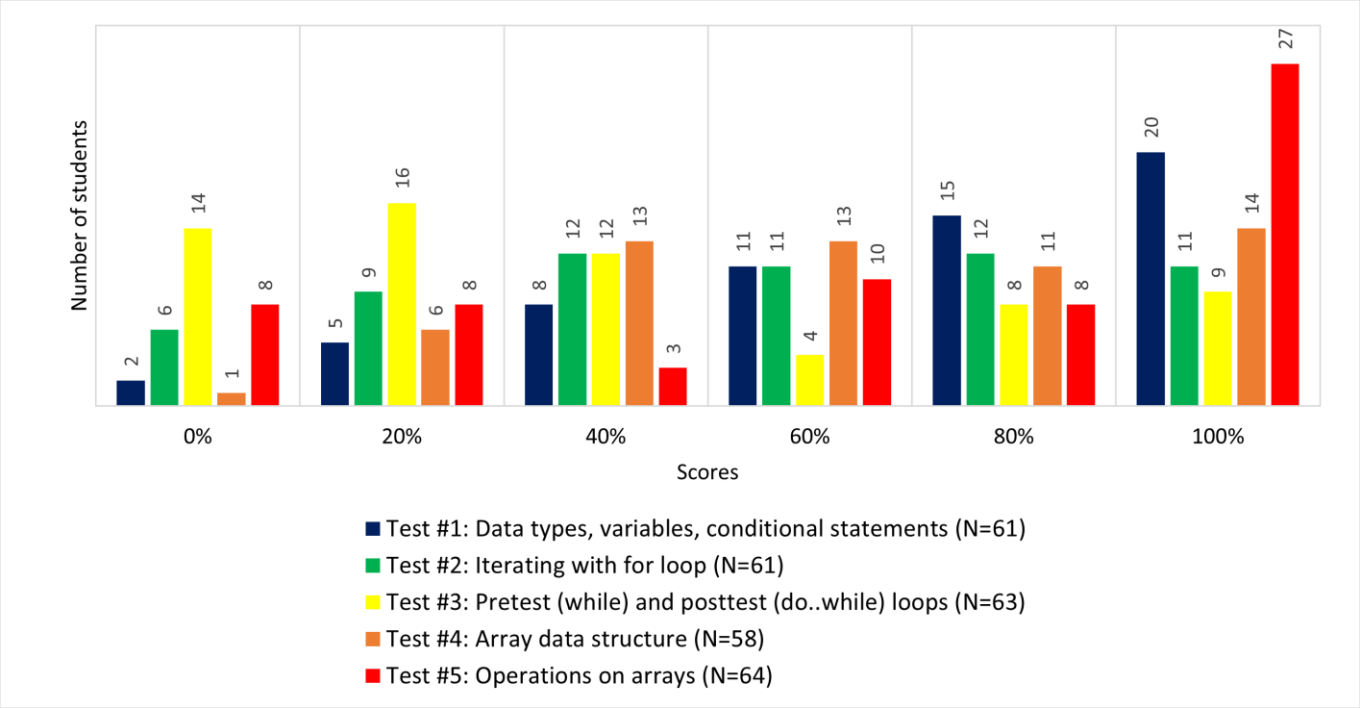

Fig. 2 Distribution of students' scores in the PR1 course tests

In Fig. 3, we observe that the most challenging task in the PR2 course was test #7, which concentrated on recursion and the development of simple recursive functions. Most students scored only 20% or 40% on this test. In contrast, the most straightforward test was test #10, which dealt with operations on arrays, where most students scored 60%, 80%, or 100%.

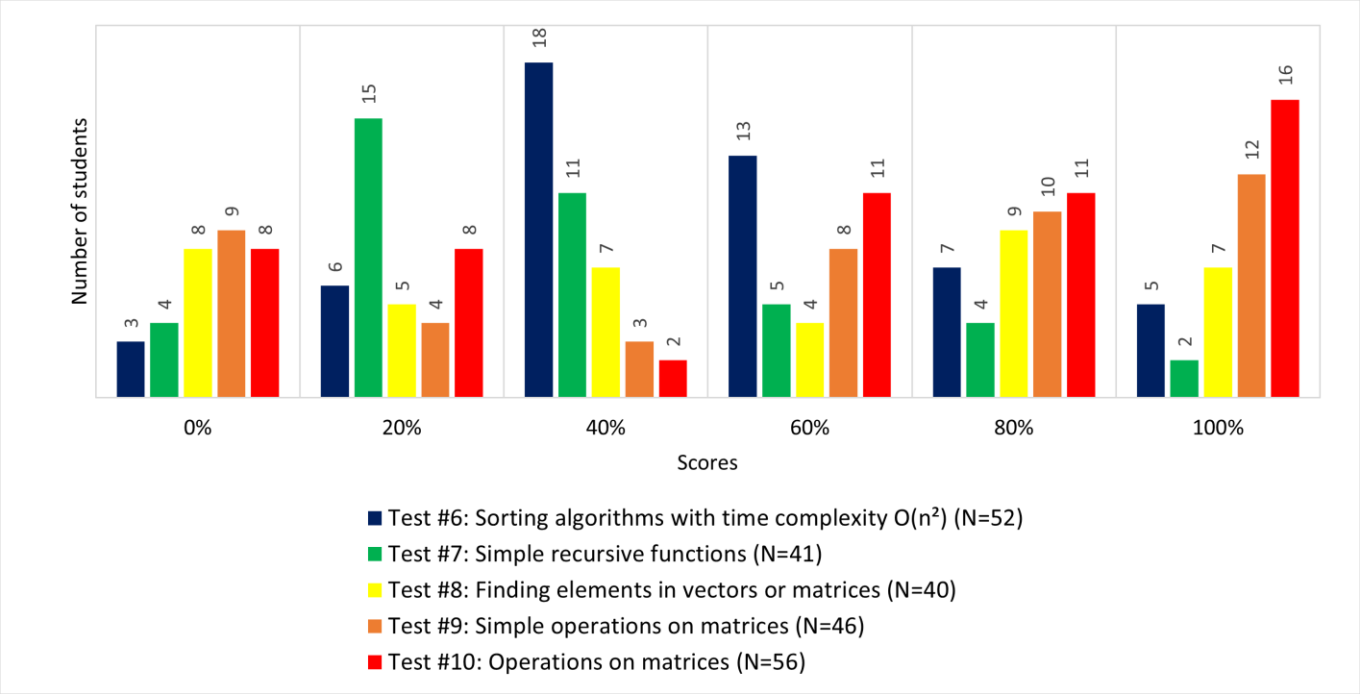

Fig. 3 Distribution of students' scores in the PR2 course tests

To better understand the relationships between topics in the PR1 and PR2 programming courses, we calculated the correlation matrices of the test scores. The results are depicted in Fig. 4.

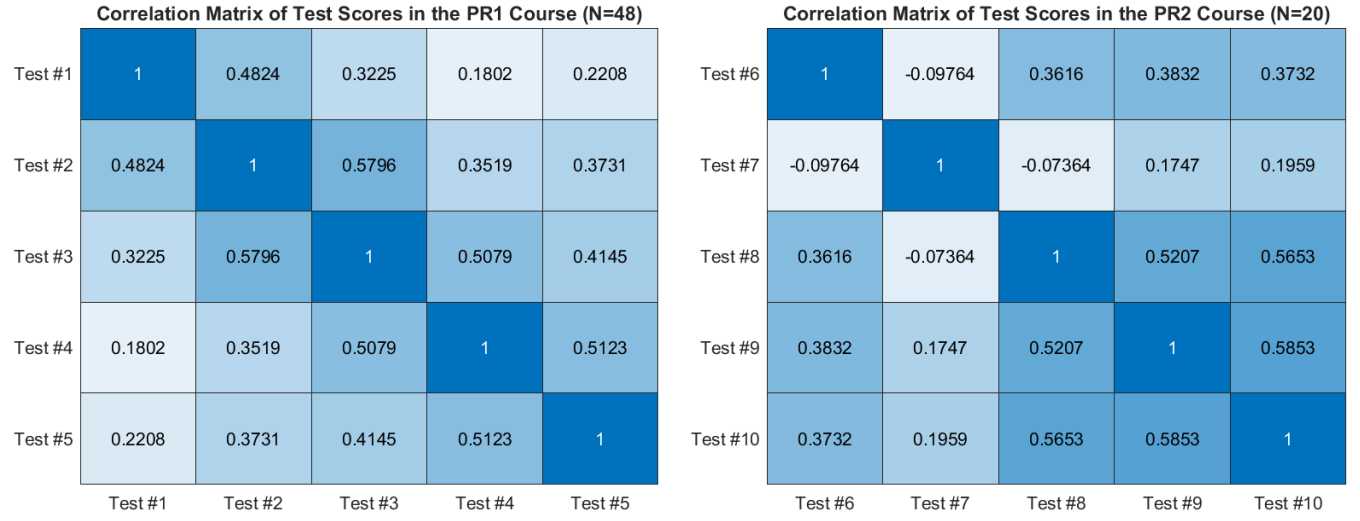

Fig. 4 Correlation matrices showing the relationships between test scores for the PR1 and PR2 courses

The correlation matrix of test scores in the PR1 course indicates that the subsequent topics are closely related. This relationship is understandable, as the curriculum is designed for the topics to build upon one another in this first introductory computer programming course.

The correlation matrix of scores in PR2 is somewhat surprising. There is no correlation between test #6 and test #7, as well as between test #7 and test #8. This lack of correlation may be attributed to the differing focuses of the tests. Test #7 emphasized the development of simple recursive functions, while test #6 concentrated on sorting algorithms, specifically simple exchange sort, bubble sort, insertion sort, minimum selection sort, and maximum selection sort. Test #8, on the other hand, dealt with searching algorithms for vectors and matrices. During test #7, students were required to engage in logical thinking to understand recursion and develop recursive functions. In contrast, tests #6 and #8 required students to understand and memorize specific algorithms. The correlation matrix of scores in PR2 also indicates that the most closely related tests were test #8, test #9, and test #10. This makes sense, as all three tests

focused on different algorithms involving vectors and matrices, as well as various operations on matrices. Therefore, the topics are interconnected.

V. DISCUSSION

The use of Moodle-based assessments featuring randomized questions has proven to be advantageous for both students and instructors. This approach allows for more personalized assessments, decreasing the chances of students copying answers and providing a more accurate reflection of their individual understanding. Additionally, the automated grading system in Moodle greatly reduces the workload for instructors, making it easier to provide timely feedback. These advantages are consistent with previous studies highlighting the benefits of LMS in programming education [12], [13].

The study's results highlight the difficulties students encounter when trying to master specific programming concepts, especially recursion and loop structures. The performance data indicates that while students generally understand basic syntax and operations involving arrays, they struggle with more abstract problem-solving tasks, such as developing recursive functions. This finding is consistent with prior research, which shows that recursion is one of the most challenging topics for beginners due to its conceptual complexity [15].

Future research should investigate more strategies to assist students who struggle with recursion and loop structures. Adaptive learning methods, such as personalized exercises and interactive visualizations, may help close the gap for students needing additional practice.

VI. CONCLUSION

This study demonstrated the effectiveness of using Moodle-based tests with randomized questions to assess students' progress in introductory programming courses. The results highlighted key challenges that students face, particularly with recursion and loop structures, while also confirming the logical progression of topics within the curriculum. Correlation analysis provided insights into the relationships among different programming concepts, suggesting potential improvements in course structure and instructional strategies.

ACKNOWLEDGMENT

The paper was supported by the national project KEGA 014TTU-4/2024 “Intelligent Animation-Simulation Models, Tools, and Environments for Deep Learning.”

REFERENCES

[1] L. Végh, “From Bubblesort to Quicksort with Playing a Game (Hravou formou od bublinkového triedenia po rýchle triedenie),” in *XXIX International Colloquium on the Management of Educational Process, Brno, CZ*, 2011, pp. 539–549.

[2] L. Végh, “Javascript Library for Developing Interactive Micro-Level Animations for Teaching and Learning Algorithms on One-Dimensional Arrays,” *Acta Didactica Napocensia*, vol. 9, no. 2, 2016.

[3] L. Végh and V. Stoffová, “An interactive animation for learning sorting algorithms: How students reduced the number of comparisons in a sorting algorithm by playing a didactic game,” *Teaching Mathematics and Computer Science*, vol. 14, no. 1, pp. 45–62, 2016.

[4] L. Végh and V. Stoffová, “Algorithm Animations for Teaching and Learning the Main Ideas of Basic Sortings,” *Informatics in Education*, vol. 16, no. 1, pp. 121–140, Apr. 2017, doi: 10.15388/infedu.2017.07.

[5] M. K. Bagga and L. S. Agrati, “Editorial: Digitalization in education: developing tools for effective learning and personalisation of education,” *Front. Educ.*, vol. 9, p. 1463596, Aug. 2024, doi: 10.3389/feduc.2024.1463596.

[6] “Moodle and assessment,” Moodle. Accessed: Jan. 21, 2025. [Online]. Available: https://moodle.com/moodle-and-assessment/

[7] P. Etemadfar, H. Barati, and A. Dabbaghi, “The Effect of Flipped Classroom Integrated with MOOCs on Intermediate EFL Learnersrsquo; Use of Self-Regulated Learning Strategies in Writing,” *JELTL*, no. Online First, Oct. 2024, doi: 10.22034/elt.2024.63311.2689.

[8] M. Reina *et al.*, “PLATA: Design of an Online Platform for Chemistry Undergraduate Fully Automated Assignments,” *J. Chem. Educ.*, vol. 101, no. 3, pp. 1024–1035, Mar. 2024, doi: 10.1021/acs.jchemed.3c00962.

[9] M. Huerta-Gomez-Merodio and M. V. Requena-Garcia-Cruz, “Application of MS Excel and FastTest PlugIn to automatically evaluate the students’ performance in structural engineering courses,” *Comp Applic In Engineering*, vol. 32, no. 6, p. e22799, Nov. 2024, doi: 10.1002/cae.22799.
[10] S. Karkina, J. Mena, R. Valeeva, I. Yarmakeev, E. Dyganova, and M. Bhullar, “Fostering future music teachers’ professional skills: developing a signature pedagogy using e-learning,” *Front. Educ.*, vol. 8, p. 1162748, Sep. 2023, doi: 10.3389/feduc.2023.1162748.
[11] N. Popovic, T. Popovic, I. Rovcanin Dragovic, and O. Cmiljanic, “A Moodle-based blended learning solution for physiology education in Montenegro: a case study,” *Advances in Physiology Education*, vol. 42, no. 1, pp. 111–117, Mar. 2018, doi: 10.1152/advan.00155.2017.
[12] H. Hickman and T. Bell, “Automated Assessment: Does It Align With Teachers’ Views?,” in *Proceedings of the 19th WiPSCE Conference on Primary and Secondary Computing Education Research*, Munich Germany: ACM, Sep. 2024, pp. 1–10. doi: 10.1145/3677619.3678113.
[13] M. Kaya and S. A. Özel, “Integrating an online compiler and a plagiarism detection tool into the Moodle distance education system for easy assessment of programming assignments,” *Comp Applic In Engineering*, vol. 23, no. 3, pp. 363–373, May 2015, doi: 10.1002/cae.21606.
[14] S. H. P. W. Gamage, J. R. Ayres, and M. B. Behrend, “A systematic review on trends in using Moodle for teaching and learning,” *IJ STEM Ed*, vol. 9, no. 1, p. 9, Dec. 2022, doi: 10.1186/s40594-021-00323-x.
[15] J. Nzemeke, “From Misconceptions to Mastery: Addressing Novice Students’ Misconceptions in Recursive Algorithm Learning,” in *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 2*, Turku Finland: ACM, Jun. 2023, pp. 611–612. doi: 10.1145/3587103.3594143.