Jiang, B. et al. (Eds.) (2025). Proceedings of the 33rd International Conference on Computers in Education. Asia-Pacific Society for Computers in Education

Exploring the Holistic Impact of Adaptive Formative Assessment for Novice Programmers

Jagadeeswaran THANGARAJ[a*], Monica WARD[a] & Fiona O'RIORDAN[b]
[a]*School Of Computing, Dublin City University, Ireland*
[b]*Teaching and Learning Unit, CCT College Dublin, Ireland*
*Jagadeeswaran.Thangaraj2@mail.dcu.ie

Abstract: The Introductory Programming module is the first step in the software development related courses. The learning outcomes of this module are fundamental concepts of programming. These independent components increase the difficulties for novices including student struggle, and low self-confidence. To motivate novice students to learn programming languages, lecturers employ a variety of exercises. Inability to address errors while programming can lead novice students to lose interest, making the deliberate introduction of common code errors—alongside strategies to increase motivation and confidence — a key element in supporting their comprehension in programming courses. Formative assessment is one of the approaches for effective programming learning that aims to increase student understanding, instructor instruction, and learning by providing feedback on students' progress. Most research focuses on either formative assessment or adaptive learning, but not their intersection in programming. Inspired by this approach, this study proposes the use of adaptive formative assessment as a pedagogical intervention to enhance student confidence and support learning Introductory Programming. The experiment is based on lessons learned from the literature and pedagogical theories that support learning through assessment and scaffolding. This study investigated how effectively these assessments helped students understand, learn and develop a sense of their ability to develop computer programs by introducing common programming errors. The self-confidence of students to learn programming is measured through a survey questionnaire and the assessments impact. Findings showed that employing adaptive formative assessment was more likely to motivate students and increase their self-confidence.

Keywords: Adaptive assessment, AI, Computer programming, Formative feedback, Introductory programming, Novice students, Self-confidence.

1. Introduction

Novice programmers find difficulty in: grasping the fundamental concepts of program ming structure; learning programming language syntax; and identifying errors and troubleshooting program code (Luxton-Reilly et al. 2018). The assessment and feedback systems examine the programs' individual elements and point out any areas where mistakes were made by the students. Novices' confidence when learning computer programming can be increased by giving them immediate feedback (Pfitzner-Eden 2016). Formative feedback is information given to a student with the goal of changing their way of thinking or acting to enhance learning (Shute 2008). Although adaptive formative assessment is in use in the education sector, it gets more attention in recent times by computer assisted learning (Crow, Luxton-Reilly, and Wuensche 2018; Ericson, McCall, and Cunningham 2019). These allow academics to design flexible methods in engagement of learning; in particular, these models allow supporting the students with feedback to make them understand the concepts of computer sciences (Barana, Fissore, and Marchisio 2020). The objective of this research is to use adaptive formative assessment to increase students' level of confidence in learning introductory programming

and their actual performance on programming tests. The research questions that this study addresses are:

***RQ-1*:** Can an adaptive formative assessment help build self-confidence in novice programmers in learning basic concepts of programming?

***RQ-2*:** Can an adaptive formative assessment support the ability of novices to understand and correct errors and encourage them to improve their programming skills?

2. Background

2.1 Self-confidence

A number of non-cognitive elements, including self-confidence, self-efficacy, and self-awareness, can be linked to learning outcomes in the context of programming education (Aljowaed and Alebaikan 2018). Self-confidence, or the student's assessment of their own performance, is an important factor that affects how well they learn programming (Kővári and Katona 2023). Underconfident students also have challenges and don't work hard enough to solve difficult programming and algorithmization problems (Kővári and Katona 2023). Students could therefore try the exercises as many times as they wanted and get feedback. Numerous automated assessment systems and tools have been developed for aiding students and teachers (Luxton-Reilly et al. 2023). In order to increase novices' confidence in their ability to learn, this study aimed to develop formative assessments for introductory programming.

2.2 Adaptive Formative Assessment

Adaptive formative assessments are customized to each student individually based on their responses to previous test items (Papanastasiou 2021). Every student works on a customized set of tasks since the questions are chosen by an algorithm that considers previously provided answers. Therefore, it varies from traditional assessment in that each participant is asked a separate set of questions rather than all the same ones (Vie, Popineau, Bruillard, and Bourda 2017). Using adaptive assessment, which organizes a collection of questions into three cognitive levels according to complexity (easy, moderate, and difficult), programming adaptive testing evaluates students' knowledge in programming courses (Chatzopoulou and Economides 2010). Two characteristics were deemed crucial while intending to develop adaptive formative assessments of programming tasks. For error messages to be properly understood, feedback needs to be quick and detailed. Second, students must be given the chance to discover their mistakes after making a number of at tempts on different questions. Due to the nature of these elements, it is necessary to create assessments large enough so that students may repeat assessments without encountering the same questions twice. This paper presents how to create an adaptive formative assessment in an efficient manner so that students can grasp the material and get a motivation of how proficient they are with computer programming. While considering the limitations of automatic assessment, this framework emphasizes the importance of achieving comparable difficulty for questions and maximizes the potential for randomization for exercise tasks. This helps to create meaningful feedback for students and supports their learning process.

This study introduces a novel adaptive formative assessment framework that lever ages common programming errors as structured learning opportunities within a difficulty based progression model. Unlike existing adaptive systems that adjust solely based on correctness, our approach dynamically adjusts question difficulty according to both the type and recurrence of errors made by the learner. Error patterns are deliberately embedded and scaffolded across low-, medium-, and high-difficulty levels, enabling students to progressively develop debugging skills while reinforcing core programming concepts. By integrating customised feedback, the system provides a personalised, iterative learning pathway that promotes both conceptual mastery and self-confidence in novice programmers.

2.3 Methodology

The University's first year 'Introductory programming' module provided the data for this investigation. The data includes 77 students' programming quiz attempts that they turned in at the end of each quiz session. In proportion to the number of quiz attempts, some students attempted numerous surveys. These quizzes were conducted periodically during teaching sessions to build novices' confidence as well as to capture their barriers in programming. The respondents were questioned about how they felt about adaptive formative assessment quizzes of each programming topic using Likert scale survey questions.

3. Results

RQ-1: Can adaptive formative assessment help build self-confidence in novice programmers in learning basic concepts of programming?

We questioned about four subjects including their level of confidence in learning computer programming, designing new programs, understanding how programs operate, and understanding programming errors. Their confidence levels have significantly raised, as seen by the mean difference between their pre- and post-quiz tries. Before the quizzes, the confidence scores ranged from 0 to 10, and after the quizzes, they rose to between 2 and 10. To determine the impact of adaptive formative assessment in these subjects, a paired-sample *T-test* was used. There was a statistically significant difference between the pre-quiz (M = 4.63 out of 10, SD = 2.212) and the post-quiz (M = 6.23 out of 10, SD = 1.593) for learning computer programming ($t[62] = 4.832$, $p < 0.001$ [two-tailed]). Their confidence levels before and after the quiz exercises differed statistically significantly in all four elements.

RQ-2: Can adaptive formative assessment support the ability of novices in understanding and correcting errors and encouraging them to improve their programming skills?

Three questions about their comprehension of errors and how to correct them were asked in this study in order to answer RQ-2. Following each quiz, a total of 246 responses were obtained. The responses ranged from 'Strongly disagree'(1) to 'Strongly agree'(5). According to One-Sample *T*-test, the 95% confidence interval falls between 3.30 and 3.56 for understanding common code errors. Also it falls between 3.28 & 3.55 and 3.29 & 3.56 to correct the common errors and increase the confidence in recognizing and fixing common code errors. These findings indicate that the adaptive formative assessment quizzes improved their understanding of common code errors as *mean* value is between 3 and 4. These outcomes show that the quizzes aids in their understanding of the frequent errors of programming.

4. Discussion

It investigates the impact of adaptive formative assessment over novices self-confidence on their comprehension of fundamental programming concepts and their capacity to recognize and correct common errors in programming after attending the quizzes. Findings from students' self-rated surveys and analysis techniques, novice students' self-confidence and comprehension of independent programming concepts have significantly increased as a result of the use of difficulty-based adaptive strategies and the introduction of numerous errors in a formative assessment. As a result, learning opportunities have expanded, increasing students' confidence, and understanding the common code errors. Because the questions in the evaluation system are only shown dependent on the responses to earlier questions. Therefore, proficient learners do not require more time. It can be a viable teaching and learning tool for introductory programming.

2.3 Future work

Creating enhanced error messages from compiler messages is challenging and it requires more work to generate the enhanced error messages. Recent advancements in Generative Artificial Intelligence (Gen-AI) have made it possible to generate such messages for easier comprehension in addition to generating questions with different difficulty levels (Koutcheme and Hellas. 2024). Also, this study recommends using GenAI to categorize the questions based on their degree of difficulty. Additionally, we intend to look into additional research topics, such as skills gap analysis, to determine students' programming learning paths and difficult topics.

References

Aljowaed, M., & Alebaikan, R. A. (2018). Training needs for computer teachers to use and teach computational thinking skills. *International Journal for Research in Education*, *42*(3), 237–284.

Barana, A., Fissore, C., & Marchisio, M. (2020, 01). From standardized assessment to automatic formative assessment for adaptive teaching. In (p. 285-296). doi: 10.5220/ 0009577302850296

Chatzopoulou, D., & Economides, A. (2010, 08). Adaptive assessment of student's knowledge in programming courses. *J. Comp. Assisted Learning*, *26*, 258-269. doi: 10.1111/j.1365-2729.2010.00363.x

Crow, T., Luxton-Reilly, A., & Wuensche, B. (2018). Intelligent tutoring systems for programming education: a systematic review. In *Proceedings of the 20th australasian computing education conference* (p. 53–62). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3160489.3160492

Ericson, B., McCall, A., & Cunningham, K. (2019). Investigating the affect and effect of adaptive parsons problems. In *Proceedings of the 19th koli calling international conference on computing education research.* New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3364510.3364524

Koutcheme, C., & Hellas, A. (2024). Propagating Large Language Models Programming Feedback. Proceedings of the Eleventh ACM Conference on Learning @ Scale, 366–370. https://doi.org/10.1145/3657604.3664665

Kovari, A., & Katona, J. (2023, February). Effect of software development course on programming self-efficacy. *Education and Information Technologies*, *28*(9), 10937–10963. doi: 10 .1007/s10639-023-11617-8

Luxton-Reilly, A., Simon, Albluwi, I., Becker, B. A., Giannakos, M., Kumar, A. N., . . . Szabo, C. (2018). Introductory programming: a systematic literature review. In *Proceedings companion of the 23rd annual acm conference on innovation and technology in computer science education* (p. 55–106). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3293881.3295779

Luxton-Reilly, A., Tempero, E., Arachchilage, N., Chang, A., Denny, P., Fowler, A., . . . Ye, X. (2023). Automated assessment: Experiences from the trenches. In *Proceedings of the 25th australasian computing education conference* (p. 1–10). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3576123.3576124

Papanastasiou, E. (2021). Adaptive assessment. In R. Gunstone (Ed.), *Encyclopaedia of science education* (pp. 1–2). Dordrecht: Springer Netherlands. doi: 10.1007/978-94-007-6165-0_3-4

Pfitzner-Eden, F. (2016). Why do I feel more confident? bandura's sources predict preservice teachers' latent changes in teacher self-efficacy. *Frontiers in Psychology*, *Volume 7 - 2016*. Retrieved from https://www.frontiersin.org/journals/ psychology/articles/10.3389/fpsyg.2016.01486

Shute, V. J. (2008). Focus on Formative Feedback. *Review of Educational Research*, *78*(1), 153-189. doi: 10.3102/0034654307313795

Vie, J.-J., Popineau, F., Bruillard, , & Bourda, Y. (2017, 02). A review of recent advances in adaptive assessment. In (Vol. 94, p. 113-142). doi: 10.1007/978-3-319-52977-6_4