2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC

Host Sponsor UNC Charlotte

Paper ID #50794

The Essential Re-evaluation of Programming Education in the Age of Artificial Intelligence

Dr. Nader Kesserwan, Robert Morris University

Dr. Nader Kesserwan is an Associate Professor in the Department of Engineering at Robert Morris University in Pittsburgh, Pennsylvania. He teaches undergraduate and graduate courses in computer science, software engineering, interface design, programming languages, and project engineering.

Dr. Kesserwan brings several years of industry experience in quality assurance to his academic role, bridging practical expertise with theoretical foundations in the classroom. His research focuses on model-based testing, Internet of Things (IoT) security, wireless sensor networks, and broader areas of software engineering. He has contributed to advancing methods that improve software reliability, system security, and engineering practices.

Dr. Jameela Al-Jaroodi

Jameela Al-Jaroodi is a professor and coordinator of the software engineering undergraduate program in the Department of engineering at Robert Morris University, Pittsburgh, PA, USA. She is also the coordinator of the master of science program in engineering management. She holds a Ph.D. degree in Computer Science from the University of Nebraska-Lincoln and a M.Ed. in higher education management from the University of Pittsburgh. She is working on research in software engineering, middleware, distributed systems, and smart system with focus on Industry 4.0, Healthcare 4.0, and smart cities. In addition, she works on education/pedagogy research and outreach projects within the STEM areas.

Talal Abdullah Bin Jathlan, Robert Morris University
Faisal Hussain Alfaifi, Robert Morris University
Abdulrahman Ibrahim Alharbi

The Essential Re-evaluation of Programming Education in the Age of Artificial Intelligence

Abstract

This study critically investigates the evolving effectiveness of teaching programming languages within contemporary engineering education settings when students leverage advanced Artificial Intelligence (AI) tools. The increasing integration of AI in student coursework presents complex challenges, notably concerning academic integrity, the authenticity of submitted work, and maintaining instructional control over the learning process. Therefore, this research urgently highlights the necessity for a comprehensive re-evaluation of established teaching methodologies and traditional assessment criteria in programming education.

Employing a robust mixed-methods approach, our investigation draws upon insights from a thorough literature review, student surveys exploring perceptions and usage patterns, and in-depth analyses of student performances, comparing outcomes both with and without AI tool usage. The investigation revealed that students utilizing AI tools demonstrated significantly better outcomes, achieving improved efficiency and notably shorter coursework completion times, primarily by overcoming challenges related to syntax errors, debugging, and general inefficiency.

However, this sharp performance disparity raises profound concerns about genuine skill development and academic honesty, as students themselves expressed apprehension about over-reliance on AI, fearing a potential reduction in their critical thinking and independent problem-solving abilities. The data also revealed a widespread ambiguity among the student population regarding how AI-generated code intersects with existing definitions of plagiarism, underscoring the urgent need for clear ethical guidance.

The findings collectively argue for a fundamental shift in current teaching methodologies, moving beyond simple prohibition to embrace a model of AI literacy and ethical collaboration. We propose that educators integrate AI literacy into the curriculum and redesign assessments toward formats that are either AI-resistant—such as live coding sessions—or AI-integrated, requiring critical review and justification of AI-generated outputs. The ultimate goal is to enhance academic outcomes and foster genuine skill development by ensuring that students are prepared to understand, leverage, and ethically navigate the complexities of an increasingly AI-driven professional landscape.

Keywords: Academic Integrity, AI in Education, Programming Education, Teaching Methodologies, Student Assessment, Control Challenges.

1. Introduction

Programming education has traditionally relied on structured assignments designed to evaluate individual comprehension and problem-solving ability. However, the widespread availability of generative AI tools such as ChatGPT and AI-enabled development environments has altered how students approach programming tasks. Rather than prohibiting AI use outright, educators are increasingly confronted with a practical question: How should programming instruction adapt when AI tools are readily available to students? This paper presents classroom-based observations and student feedback to inform instructional practice in this evolving context.

2. Literature Review

Advances in AI are transforming many sectors, including education. In programming instruction, AI tools are playing an increasingly central role, influencing teaching methods and student learning. This review synthesizes research on the use of AI in programming education, highlighting its benefits, the challenges it introduces, and its impact on teaching and assessment.

AI tools—including intelligent tutoring systems, code generators (e.g., GitHub Copilot, ChatGPT), and adaptive learning platforms—offer notable enhancements to programming instruction (Liu, 2024). Their primary contribution

is personalized learning: AI can track student progress, identify knowledge gaps, and tailor content and feedback accordingly (Akintola et al., 2023), improving engagement and learning outcomes (Bhardwaj et al., 2025).

AI also strengthens assessment and feedback through automated grading, timely responses, and detailed performance analytics (Vetrivel et al., 2025). These systems detect common errors, guide debugging, and support interactive learning cycles (McCauley et al., 2008). Early findings show faster task completion and improved performance with AI assistance, though gains in deeper understanding remain uncertain (Alanazi et al., 2025). Students generally view AI tools as helpful for efficiency and proficiency (Kohen-Vacs et al., 2025).

Despite these benefits, AI's accessibility raises concerns about academic integrity (Pedro et al., 2019). Tools like ChatGPT can produce complete solutions, making it harder to judge authorship and actual student comprehension (Atlas, 2023) and challenging traditional definitions of plagiarism (Ogwueleka, 2025).

Educators must balance restricting AI—risking misalignment with industry practice—and permitting it, which may encourage superficial learning (Nwozor, 2025). Students also worry about overreliance, reduced critical thinking, and AI inaccuracies (Zhai et al., 2024). These challenges highlight the need for clearer institutional policies and ethical guidelines (Nguyen et al., 2023).

AI integration requires rethinking programming pedagogy and assessment (Pedro et al., 2019). Traditional assignments and exams may no longer reflect genuine learning when AI can solve them effectively (Stanoyevitch, 2024). Instruction should shift toward deeper reasoning, debugging, and effective human–AI collaboration (George, 2023).

Emerging strategies include AI-resistant assessments—live coding, code modification, algorithm design, and oral explanations (Green, 2025)—and AI-integrated tasks, such as portfolios documenting AI use or reviews of AI-generated code (Alenezi et al., 2025). These approaches emphasize conceptual understanding and informed use of AI tools (Younis et al., 2024), aligning instruction with industry expectations of human–AI partnership (Luckin, 2025).

Although research on AI in education is expanding, comparative studies examining programming performance with and without AI tools remain limited (Zawacki-Richter et al., 2022). More evidence is needed to guide effective teaching and assessment in AI-supported environments.

This study addresses these gaps by examining AI's impact on programming instruction through literature analysis, student survey data, and performance comparisons. The findings aim to inform teaching methods and assessment criteria that improve outcomes while addressing issues of assessment reliability, instructional effectiveness, and academic integrity.

3. Study Framework

This research adopts a mixed-methods framework to examine how AI tools influence programming learning. Quantitative evidence was gathered from student performance comparisons and survey responses, while qualitative insights were drawn from open-ended feedback and existing literature.

3.1 Research Design

This study was conducted in undergraduate programming courses within Computer Science, Cybersecurity, and Software Engineering programs. Two instructional conditions were observed: AI-Permitted Assignments (students were allowed to use AI tools with ethical-use guidance) and AI-Restricted Assignments (students completed comparable coursework without AI access). A total of 99 anonymized programming submissions were analyzed, and 99 students completed a survey on AI usage and perceptions. Assignments covered foundational programming concepts such as control structures, object-oriented programming, data structures, debugging, and algorithm design.

3.2 Analytical Procedures

Survey responses were analyzed using descriptive statistics and thematic coding for open-ended items. Coursework data were statistically compared across AI and non-AI conditions to identify differences in performance and potential links to AI-detection indicators. Literature findings were used to support and interpret the empirical results.

All participants provided consent, and data were collected anonymously or anonymized before analysis. Storage and access adhered to secure university protocols. All procedures complied with institutional and federal guidelines for research involving human participants.

4. Student Performance Observations

This section presents key findings from the student survey and performance analysis, addressing the impact of AI tools on programming education and associated challenges.

Performance Without AI: 25.93% earned B or higher, 18.52% failed, and students reported longer debugging time and frustration.

Performance With AI Permitted: 50.93% earned B or higher, 0% failed, and students completed assignments significantly faster.

A statistical comparison indicated a significant difference ($p < 0.01$) between the two conditions. AI tools dramatically improved syntax correction, debugging speed, and code generation efficiency. However, faster completion did not necessarily indicate deeper conceptual understanding.

4.1 Student Perception of AI

Students reported faster problem-solving, reduced frustration, clearer explanations, and increased confidence when debugging. However, concerns included over-reliance, reduced independent problem-solving, and ambiguity regarding plagiarism. When asked whether AI-generated code increases plagiarism risk: 44% said Yes, 46% were Not Sure, and 10% said No. When asked about restrictions: 74% supported limited restrictions, 24% preferred no restrictions, and 2% supported full restrictions.

4.2 Instructional Challenges Identified

Without AI access, students reported syntax errors, debugging difficulty, time inefficiency, lack of immediate feedback, and difficulty understanding complex concepts. These findings suggest that AI tools are filling instructional gaps traditionally addressed through instructor feedback and office hours.

4.3 Pedagogical Implications for Engineering Education

Programming instruction should shift from a binary 'allow vs prohibit' approach toward structured integration.

Teach AI Literacy: Students must learn when to use AI, how to verify AI output, how to critically evaluate generated code, and ethical boundaries of AI usage.

Redesign Assessments: AI-Resistant assessments such as in-class coding exercises, live debugging sessions, and oral code explanations. AI-Integrated assessments requiring documentation of AI usage, critical review of AI-generated code, and reflection on AI-assisted decisions.
Emphasize Process Over Product: Instructors should evaluate student reasoning, design decisions, and ability to modify and explain code, not merely final correctness.

5. Discussion

For engineering educators, AI is becoming a permanent component of professional practice. The question is not whether students will use AI, but how programming education can be structured so AI enhances rather than replaces learning. This requires transparent academic integrity policies, assessment redesign, faculty development in AI pedagogy, and institutional clarity on ethical use.

Implications for teaching: Curricula should integrate AI literacy, teaching ethical and effective collaboration with AI tools. Assessments require redesign:

- **AI-Resistant:** In-person coding challenges, live debugging, and code explanations to evaluate independent skills.
- **AI-Integrated:** Assignments using AI with required documentation and critical review, focusing on process and reasoning.

Overall, AI demonstrates transformative potential in programming education, enhancing efficiency and learning outcomes while necessitating structured integration, responsible use, and clear assessment and policy frameworks.

6. Recommendations and Solutions

Our findings highlight the transformative yet challenging role of AI in programming education. To harness AI's benefits while mitigating risks to academic integrity and core skill development, we propose the following:

6.1 Teaching Methodologies for an AI-Integrated Classroom

Educators should treat AI as a collaborative learning aid rather than a shortcut, modeling responsible use for debugging, conceptual understanding, and brainstorming. Curricula should shift toward higher-order thinking, emphasizing algorithmic design, complex problem-solving, and critical analysis of AI-generated code through live coding, code reviews, and explain-your-code assignments. Integrating AI literacy—covering tool capabilities, limitations, biases, and ethics—prepares students for professional human-AI collaboration.

6.2 Assessment Strategies

Given students' uncertainty about AI use and plagiarism, assessments should be redesigned to ensure genuine learning. Recommendations include in-class proctored exams, iterative project-based submissions with documented AI usage, and personalized tasks that challenge generic AI solutions. Clear AI usage policies should define acceptable practices and emphasize the learning process alongside final outputs. AI detection tools should support feedback and learning rather than serving purely punitive purposes.

6.3 Support and Resources

To reduce over-reliance on AI, institutions should provide structured debugging workshops, curated AI tutorials, and guidance on effective AI use. Promoting human-to-human collaboration encourages deeper learning through inquiry, peer discussion, and instructor support.

7. Study Contributions and Limitations

This study sheds light on AI's impact in programming education within a real-world classroom setting at Robert Morris University. Using a quasi-experimental design, we observed meaningful performance differences, though causal claims should be interpreted cautiously due to potential pre-existing group differences. Student surveys provided valuable self-reported perspectives, and future research could incorporate additional data collection to reduce bias. While fully quantifying all AI interactions was beyond this study's scope, the inclusion of AI detection scores offered useful insights and highlights the evolving nature of AI assessment. Overall, the findings provide a foundation for further research exploring broader applicability and methodological refinement.

8. Conclusion and Future Work

This study demonstrates that AI tools can significantly enhance programming education by improving performance and efficiency, while also raising concerns about academic integrity and depth of learning. The results emphasize the need for educators to adapt teaching methods and assessment strategies to integrate AI responsibly as a learning aid.

Future research should focus on longitudinal studies to assess long-term skill retention, controlled experiments with larger and diverse samples, and the development of AI-integrated pedagogical interventions. Investigating specific AI interactions that promote learning versus dependency, and exploring detailed perceptions of academic integrity through qualitative methods, will further inform best practices. Addressing these areas can help institutions foster equitable, effective, and ethical learning environments, preparing students to use AI thoughtfully in professional contexts.

References

1. Ciarli, T., Kenney, M., Massini, S., & Piscitello, L. (2021). Digital technologies, innovation, and skills: Emerging trajectories and challenges. Research Policy, 50(7), 104289.
2. Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code. org. Computers in Human Behavior, 52, 200-210.
3. Linn, M. C., & Dalbey, J. (1985). Cognitive consequences of programming instruction: Instruction, access, and ability. Educational Psychologist, 20(4), 191-206.
4. Philbin, C. A. (2023). Exploring the potential of artificial intelligence program generators in computer programming education for students. ACM Inroads, 14(3), 30-38.
5. Tariq, M. U. (2024). Navigating the personalization pathway: Implementing adaptive learning technologies in higher education. In Adaptive Learning Technologies for Higher Education (pp. 265-291). IGI Global.
6. Hooda, M., Rana, C., Dahiya, O., Rizwan, A., & Hossain, M. S. (2022). Artificial intelligence for assessment and feedback to enhance student success in higher education. Mathematical Problems in Engineering, 2022(1), 5215722.
7. Sullivan, M., Kelly, A., & McLaughlan, P. (2023). ChatGPT in higher education: Considerations for academic integrity and student learning. Journal of Applied Learning & Teaching, 6(1), 31-40.
8. Liu, J., & Li, S. (2024). Toward artificial intelligence-human paired programming: A review of the educational applications and research on artificial intelligence code-generation tools. Journal of Educational Computing Research, 62(5), 1165-1195.
9. Akintola, A. S., Akintayo, M., Kadri, T., Oforgu, C., Michael, M., & Nwanna, M. (2025). Adaptive AI systems in education: real-time personalised learning pathways for skill development. J. Artif. Intell. Mach. Learn. Data Sci., 3, 2489-2494.
10. Bhardwaj, V., Zhang, S., Tan, Y. Q., & Pandey, V. (2025, February). Redefining learning: student-centered strategies for academic and personal growth. In Frontiers in Education (Vol. 10, p. 1518602). Frontiers Media SA.
11. Vetrivel, S. C., Arun, V. P., Ambikapathi, R., & Saravanan, T. P. (2025). Automated Grading Systems: Enhancing Efficiency and Consistency in Student Assessments. In Adopting Artificial Intelligence Tools in Higher Education (pp. 41-61). CRC Press.
12. McCauley, R., Fitzgerald, S., Lewandowski, G., Murphy, L., Simon, B., Thomas, L., & Zander, C. (2008). Debugging: a review of the literature from an educational perspective. Computer Science Education, 18(2), 67-92.
13. Alanazi, M., Soh, B., Samra, H., & Li, A. (2025). The Influence of Artificial Intelligence Tools on Learning Outcomes in Computer Programming: A Systematic Review and Meta-Analysis. Computers (2073-431X), 14(5).
14. Kohen-Vacs, D., Usher, M., & Jansen, M. (2025). Integrating generative AI into programming education: Student perceptions and the challenge of correcting AI errors. International Journal of Artificial Intelligence in Education, 1-19.
15. Pedro, F., Subosa, M., Rivas, A., & Valverde, P. (2019). Artificial intelligence in education: Challenges and opportunities for sustainable development.

16. Atlas, S. (2023). ChatGPT for higher education and professional development: A guide to conversational AI.
17. Ogwueleka, F. N. (2025). Plagiarism Detection in the age of artificial intelligence: current technologies and future directions. AI and ethics, academic integrity and the future of quality assurance in higher education, 10.
18. Nwozor, A. (2025). Artificial intelligence (AI) and academic honesty-dishonesty nexus: Trends and preventive measures. *AI and Ethics, Academic Integrity and the Future of Quality Assurance in Higher Education*, *27*.
19. Zhai, C., Wibowo, S., & Li, L. D. (2024). The effects of over-reliance on AI dialogue systems on students' cognitive abilities: a systematic review. *Smart Learning Environments*, *11*(1), 28.
20. Nguyen, A., Ngo, H. N., Hong, Y., Dang, B., & Nguyen, B. P. T. (2023). Ethical principles for artificial intelligence in education. Education and information technologies, 28(4), 4221-4241.
21. Pedro, F., Subosa, M., Rivas, A., & Valverde, P. (2019). Artificial intelligence in education: Challenges and opportunities for sustainable development.
22. Stanoyevitch, A. (2024). Online assessment in the age of artificial intelligence. Discover Education, 3(1), 126.
23. George, A. S. (2023). Preparing students for an AI-driven world: Rethinking curriculum and pedagogy in the age of artificial intelligence. Partners Universal Innovative Research Publication, 1(2), 112-136.
24. Green, A. (2025). An Analysis of LLM Use in Introductory Programming Education and Development of AI Resistant Assessments via Code Reviews (Master's thesis, Purdue University).
25. Alenezi, M., & Akour, M. (2025). Ai-driven innovations in software engineering: a review of current practices and future directions. Applied Sciences, 15(3), 1344.
26. Younis, H. A., Eisa, T. A. E., Nasser, M., Sahib, T. M., Noor, A. A., Alyasiri, O. M.,. & Younis, H. A. (2024). A systematic review and meta-analysis of artificial intelligence tools in medicine and healthcare: applications, considerations, limitations, motivation and challenges. Diagnostics, 14(1), 109.
27. Luckin, R. (2025). Nurturing human intelligence in the age of AI: rethinking education for the future. Development and Learning in Organizations: An International Journal, 39(1), 1-4.
28. Zawacki-Richter, O., Marín, V. I., Bond, M., & Gouverneur, F. (2019). Systematic review of research on artificial intelligence applications in higher education–where are the educators?. International journal of educational technology in higher education, 16(1), 1-27.