International Journal of Electrical Engineering and Sustainability (IJEES)
ISSN (online): 2959-9229
https://ijees.org/index.php/ijees/index
Arab Impact Factor: 1.51 Dolj 2026 = Q1
Volume 4 | Number 2 | July-December 2026 | Pages 144-152

Recent Trends in Artificial Intelligence in Computer Programming Education

Wisam Algsaier*
Department of Computer Technologies, Higher Institute of Technical Sciences, Misrata, Libya

Email: wisamelgsaier@gmail.com

Received: May 23, 2026 Accepted: August 25, 2026 Published: September 01, 2026
DOI: 10.65998/ijees.v4i2.183

Abstract: Artificial intelligence (AI) is transforming computer programming education by introducing new methods for instruction, assessment, feedback, and personalized learning. This article reviews recent trends in the application of AI to programming education, focusing on AI-powered coding assistants, intelligent tutoring systems, automated assessment tools, adaptive learning platforms, code-generation models, and personalized feedback systems. The study evidence indicates that AI tools can improve access to immediate feedback, support debugging and code comprehension, increase student engagement, and enable more individualized learning experiences. They can also assist instructors by automating routine assessment tasks, identifying common misconceptions, and supporting the preparation of instructional materials. Nevertheless, the integration of AI raises important concerns regarding excessive dependence on automated solutions, inaccurate code generation, plagiarism, academic integrity, algorithmic bias, data privacy, and unequal access to advanced technologies. The article concludes that AI should be implemented as a complementary educational resource rather than a replacement for instructors or independent programming practice. Effective adoption requires appropriate pedagogical frameworks, transparent assessment policies, AI literacy, and continuous evaluation to ensure that these technologies strengthen computational thinking, creativity, and problem-solving skills.

Keywords: Artificial intelligence; programming education; generative AI; coding assistants; intelligent tutoring systems; automated assessment; personalized learning.

1. Introduction

Artificial Intelligence (AI) is an emerging technology with a transformative impact on education that is changing the way people learn, teach and assess. With the help of machine learning, natural language processing, intelligent tutoring systems and generative AI, learning environments have become personalized, interactive and adaptive spaces, catering to the needs of each student [1]. These technologies also hold unique importance in computer programming education because students may require immediate feedback, customization and hands-on assistance to grasp programming concepts, identify syntax errors, and develop problem solving skills [2]. All these issues can be addressed by the help of AI platforms like providing examples of code, explaining complex algorithms, finding errors, and recommending learning activities based on individual student's performance etc. [3].

With the advancement of automated code-assessment tools, adaptive learning platforms, conversational assistants, intelligent tutors, and generative coding tools, the use of AI in programming education has been rapidly growing [4]. These applications can be used to assist teachers in automating repetitive assessment tasks, reveal students' common difficulties, and help students find alternative solutions and get instant feedback [5]. However, the increasing use of AI also raises worries about ethical implications, over-reliance on AI tools, poor code generation, privacy issues, and a decline in

self-reliant computational thinking [6] . Thus, it is essential to adopt suitable pedagogical approaches that ensure active learning, creativity, critical thinking, and an actual programming skill set is maintained when integrating AI.

A systematic review and meta-analysis of the study outcomes of AI tools like ChatGPT 3.5 and GitHub Copilot on learning outcomes in computer programming courses in [7] have been conducted. A total of 35 controlled studies were included in the analysis from 2020-2024. The results showed that using AI-powered learning tools had a significant positive impact on student learning outcomes compared to traditional learning methods. We then reviewed the landscape of computing education research to identify the use of generative AI tools in computing education [8]. The current literature was categorized based on the topics in computing, educational contexts, and AI tools used. It also assessed the benefits and constraints of using generative AI in computing education using the indicators of performance from previous studies.

In [9], the authors described several aspects of AI literacy education in primary schools, including AI learning theories, pedagogical practices, learning materials, evaluation methods, educational outcomes and issues with implementation. The researchers employed a systematic mapping-review approach to search for empirical studies regarding primary school education and selected 25 studies for detailed analysis, using the Scopus and Web of Science databases. The authors [10] explored generative AI's impact on learning philosophically, using the extended cognition learning perspective. The study also examined the possibility they stimulate active learning or passive consumers of content produced by machines and thereby influence the acquisition of basic cognitive abilities. It reported that generative AI might reduce students' mental effort when it replaces them instead of building upon and expanding their reasoning skills.

This study provides an up-to-date and comprehensive review of the recent advances of AI in computer programming education. It identifies and categorizes AI tools and applications into coding assistants, intelligent tutoring systems, automated assessment tools, adaptive learning platforms, code-generation models, and personalized feedback systems, and assesses their impact on the programming skills, engagement, computational thinking, problem-solving, and academic performance of students. The study also explores the impact on teachers' practice and workloads, and critically considers issues surrounding inaccurate code generation, over-reliance on AI, plagiarism, academic integrity, data privacy, algorithmic bias and access inequity. The article presents a technopedagogical and ethical lens on the role of AI in programming education, along with practical implications for educators and institutions, and directions for future research on the responsible and effective use of AI in programming education.

2. Methodology

This study was performed by literature-review method and was organized by a structured method in the recent application of artificial intelligence in computer programming education. The literature search was conducted in the period 2024-2026 and the keywords were used to find relevant publications from different databases such as Scopus, Web of Science, IEEE Xplore, ScienceDirect, SpringerLink and Google Scholar, including combinations of the following: “artificial intelligence,” “programming education,” “computer programming”, “coding assistants,” “generative AI”, “intelligent tutoring systems”, “automated code assessment”, “adaptive learning”, and “personalized feedback”. The following criteria were used for the inclusion of the articles and papers: Being published in peer-reviewed journals and conference papers in English; The direct presentation of the use of AI in programming instruction, learning outcomes, assessment, feedback, and/or student engagement. Records that had been duplicated, articles that were not peer-reviewed, research articles that were not related to programming education, papers that did not have access to the full article and articles on general educational AI but not explicitly programming were excluded. The studies were categorized according to the types of AI technology, educational uses, target learners, research methods, learning outcome measured, benefits and challenges reported after title, abstract and full-text screening. The results were then thematised and compared, so as to highlight the main trends, common results, limitations of the research, ethical issues and future aspirations.

3. AI Technologies and Applications in Programming Education

The application of AI in computer programming education is increasingly prevalent and with the introduction of new techniques for teaching, testing, and personalized learning support. Teaching students to program is a difficult process, because they should learn the abstract concepts, develop logical thinking, find errors in the program and apply theory to a practical problem [11]. AI-powered tools can aid in these efforts, offering real-time feedback, automating repetitive tasks, and tailoring educational material to students' proficiency and advancement as depicted in Figure 1.

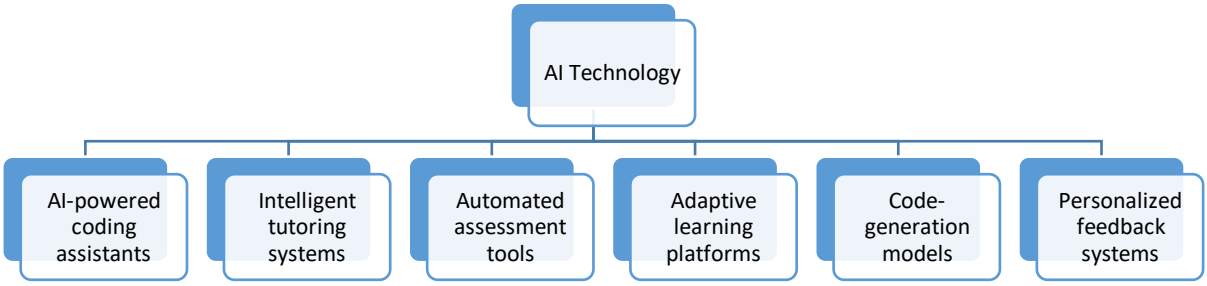

Figure 1. AI technology

The most significant use-cases in this regard are AI-powered coding assistants, intelligent tutoring systems, automated assessment systems, adaptive learning platforms, code-generation models, and personalized feedback systems. Such technologies can assist the teacher in the development of learning resources, the evaluation of assignments, the identification of learning gaps and in the focus on interventions [12]. In the meantime they provide students with real-time feedback, explore other options, and program using customisable and interactive learning experiences. Table 1 shows the list of AI Technologies and Applications in Programming Education.

Table 1. AI Technologies and Applications in Programming Education [13-18].

AI technology	Main applications	Support for programming instruction	Support for student learning
AI-powered coding assistants	Code completion, syntax correction, debugging support, code explanation, and solution recommendations.	Assist instructors in preparing demonstrations, programming exercises, and alternative solutions while reducing the time spent addressing routine coding errors.	Provide immediate assistance during coding activities, helping students identify mistakes, understand unfamiliar syntax, and explore different problem-solving approaches.
Intelligent tutoring systems	Interactive guidance, step-by-step problem solving, hints, explanations, and monitoring of learner progress.	Enable instructors to provide scalable and individualized academic support, particularly in large programming classes.	Guide students through difficult concepts at their own pace and provide targeted hints without immediately revealing complete solutions.
Automated assessment tools	Automatic code evaluation, test-case execution, error detection, plagiarism checking, and performance scoring.	Reduce the workload associated with grading programming assignments and support consistent, objective, and rapid assessment.	Give students timely feedback on code correctness, efficiency, structure, and compliance with assignment requirements.
Adaptive learning platforms	Personalized learning paths, difficulty adjustment, content recommendation, and competency tracking.	Help instructors identify knowledge gaps and organize learning materials according to students' abilities and progress.	Present exercises at an appropriate level of difficulty and allow students to progress according to their individual learning needs.

Code-generation models	Generation of programs, algorithms, functions, documentation, test cases, and code translations between programming languages.	Support the creation of instructional examples and demonstrate multiple approaches to solving the same programming problem.	Help students examine code structures, compare alternative solutions, and transform conceptual descriptions into executable programs.
Personalized feedback systems	Individualized comments, misconception detection, error explanations, performance analysis, and improvement recommendations.	Provide instructors with detailed information about common errors and individual learning difficulties, enabling targeted intervention.	Deliver specific and actionable feedback that helps students understand why an error occurred and how their code can be improved.

The use of AI technologies has a significant potential to enhance the effectiveness, accessibility, and personalisation of computer programming education. These tools can alleviate teachers' daily workload, as well as offer students timely support during their learning process when it comes to code generation, debugging, assessment, tutoring and individual feedback [19]. Their integration also may be able to provide more flexible learning spaces that are responsive to the differences in what students know, how they are doing, and how they learn.

However, the role of AI in education is determined by its responsible and pedagogically sound use. Overuse of AI-generated solutions can diminish students' independent thinking, creativity and problem-solving skills [20]. Therefore, teachers need to create learning activities that ask students to explain, assess, change and test AI-generated code. AI can play an important role in creating competent, critical, and adaptive programmers as an additional learning tool and not as a substitute for human teaching and practice [21].

4. Effects of AI on Programming Teaching and Learning

AI tools can offer real-time support to students while they are writing code, debugging, and testing, improving their programming abilities. AI Coding Assistant tools can detect code issues, clarify coding principles, provide examples, suggest alternative solutions, and help students before they even seek guidance from their teachers [22]. This fast feedback can assist learners to identify errors and change their understanding of programming structures, algorithms and software-development practices. However, the advantages of this come with the condition that the students review and test the suggestions generated by the AI before copying them, as directly copying would not give students the opportunity to develop their skills [23].

AI can also enhance the motivation and engagement of students by creating more interactive, personalized, and accessible programming activities. Errors occur repeatedly for beginners and concepts are sometimes not understood [24]. AI tutoring systems can provide personalized explanations, suggestions, and problems tailored to the student's comprehension, ensuring they learn at a suitable pace. This type of support could boost confidence and motivate ongoing involvement in programming courses [25]. However, over-reliance on AI generated answers could lead to lower persistence, productive struggle, engagement in class, and learning from failure and experimentation [26].

The impact of AI on computational thinking, problem-solving skills, and academic achievement is subtler and more complex. AI can assist students to break down programming problems, compare algorithms, analyse the efficiency of the code, and explore different ways of doing a task. These functions are present and may contribute to logical thinking and enhance the performance of assignments when they are scaffolded [27]. On the other hand, relying on AI to provide full solutions before letting students solve problems on their own could harm their ability to abstract, think algorithmically, be creative, and critically assess a problem. Better grades in such conditions can be attributed to the use of artificial intelligence rather than an actual increase in knowledge and programming skills of students [28].

Teachers can use AI to help them prepare for their lessons, create coding exercises, write illustrative examples, and give initial comments on student work [29]. Automated assessment systems may be used to assess syntax, functionality, code quality, and test results and save time for repetitive assessment tasks [30]. This allows teachers to focus more on mentoring, conceptual problem and advanced problem solving activities. However, original approaches, context or subtle misunderstandings could be overlooked when the assessment is automated [31]. But teachers may need to think again about how to assess student thinking processes, and make sure that human judgment will always be the last word in teaching and assessment, especially when it comes to materials created by AI – and the fruits of assessment.

5. Educational Benefits, Challenges, and Ethical Concerns

AI can be a powerful tool in computer programming education, offering opportunities to improve teaching and learning. AI coding assistants, intelligent tutoring systems, automated assessment tools, and personalized learning platforms can provide immediate feedback, identify coding errors, provide relevant examples, and customize learning experiences to individual needs [32]. Figure 2 depictes dimensions' of educational benefits.

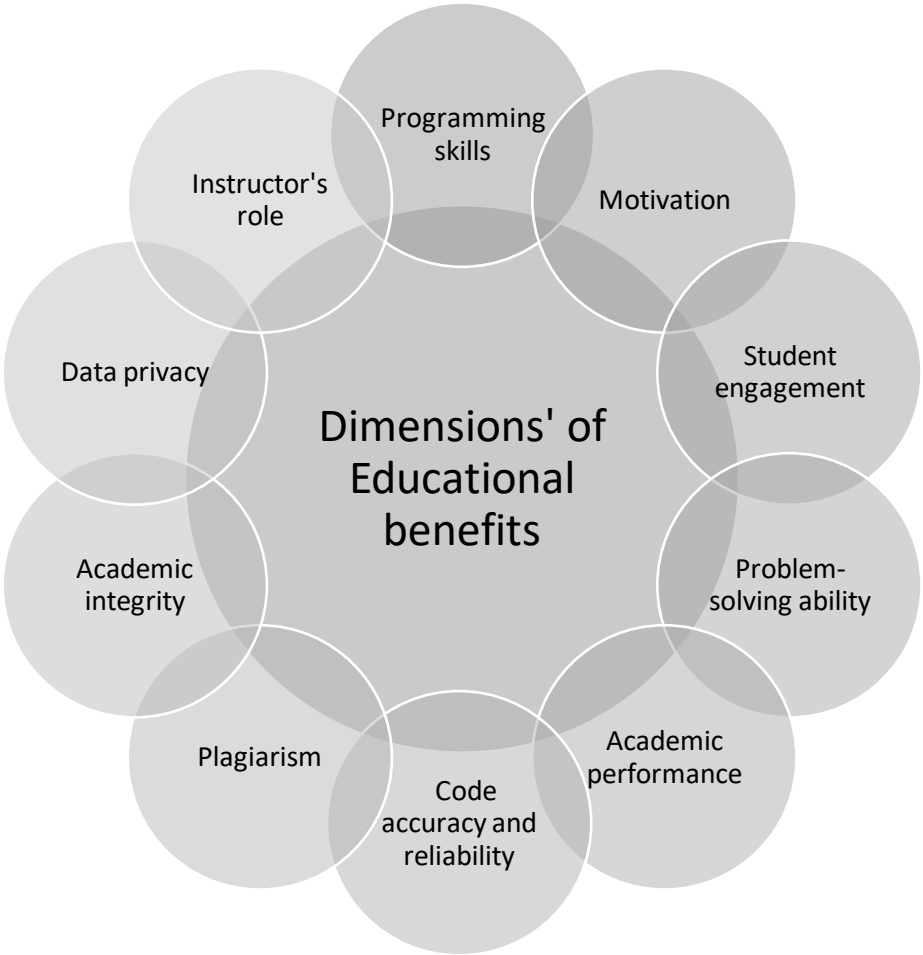

Figure 2. Dimensions' of Educational benefits.

The above mentioned functions can enhance students' program skills, motivation, engagement, problem solving skills and academic achievement. The effectiveness of these technologies in the context of education lies in how they are integrated into the curriculum, and whether students remain actively engaged in reasoning, coding, testing, and debugging processes. Table 2 thus compares the main educational advantages of AI and the technical, pedagogical and ethical issues it raises.

Table 2. Educational benefits, challenges, ethical concerns, and recommended safeguards for AI-assisted programming education [33-37].

Dimension	Educational benefits	Challenges and ethical concerns	Recommended measures
Programming skills	Provides code examples, syntax correction, debugging assistance, and immediate explanations that can accelerate skill development.	Students may copy generated code without understanding its structure, logic, or execution.	Require students to explain, modify, test, and document AI-generated code.
Motivation	Reduces frustration by offering immediate assistance and helping students overcome coding difficulties.	Easy access to complete solutions may weaken persistence and willingness to learn from errors.	Use AI for guided hints rather than complete solutions, particularly for beginners.
Student engagement	Supports interactive, personalized, and self-paced programming activities.	Frequent AI use may reduce meaningful interaction with instructors and classmates.	Combine AI-assisted activities with collaborative programming, discussion, and instructor feedback.
Problem-solving ability	Helps students compare alternative algorithms, explore solutions, and understand complex programming problems.	Excessive dependence may weaken independent reasoning, computational thinking, and creativity.	Ask students to develop an initial solution before consulting AI and compare both approaches.
Academic performance	Provides individualized support, rapid feedback, and additional practice that may improve learning outcomes.	Improved grades may reflect AI assistance rather than students' actual knowledge and abilities.	Use supervised assessments, oral examinations, practical demonstrations, and process-based evaluation.
Code accuracy and reliability	Assists in detecting syntax errors, recommending corrections, and improving code structure.	AI can generate inaccurate, insecure, inefficient, outdated, or fabricated code.	Require code testing, validation, security checks, and verification against reliable documentation.
Plagiarism	Can help students learn proper code structure and develop alternative solutions.	Direct submission of generated code may constitute plagiarism or unauthorized assistance.	Establish clear citation rules and require disclosure of how AI was used.
Academic integrity	Encourages discussion about responsible technology use and professional programming practices.	Undisclosed AI assistance can undermine assessment fairness and the credibility of academic qualifications.	Develop institutional policies defining acceptable and unacceptable uses of AI.
Data privacy	Personalized systems can analyze learning patterns and provide targeted educational support.	Student information, source code, and institutional data may be collected, stored, or exposed without adequate consent.	Use approved platforms, minimize data sharing, obtain informed consent, and comply with privacy regulations.
Instructor's role	Reduces repetitive grading and allows instructors to focus on advanced guidance and student support.	Excessive automation may reduce human judgment and overlook individual learning difficulties.	Treat AI as a supporting tool while preserving instructor responsibility and final decision-making.

AI can be a powerful enabler for the education sector, aiding in programming education, student learning, feedback generation, and skill development. However, overreliance on the use of AI could undermine independent problem solving skills, computational thinking, creativity, and critical evaluation of code. Other issues of inaccuracy in code generation, plagiarism, academic integrity, data privacy, algorithmic bias and unequal access also need to be dealt with carefully [38]. Schools could therefore have clear guidelines on appropriate AI utilization, encourage disclosure, safeguard student information, and promote equal access to approved technologies. Most importantly, AI can be used as

a complement to instructors' input, independent programming practice, human judgment and reliable assessment, not a replacement.

6. Future Directions and Conclusion

Artificial intelligence (AI) is likely to be increasingly used in programming education, particularly in more transparent, personalized, and interactive ways. Explainable AI can make it possible for students to comprehend the rationale behind a specific solution, correction, or suggestion, instead of providing code as an unexplainable answer. AI-based curricula can further tailor instruction, programming exercises, and difficulty levels based on prior knowledge, progress, and repeated mistakes of the learners. Also, using virtual reality, augmented reality or smart simulations would enable students to experience abstract programming concepts in an immersive environment, with the ability to interact with the algorithms in real-world situations. Such advances will likely foster collaborative programming between humans and AI, with students serving as the decision makers, verifiers, and problem definers while also using AI as a reasoning partner.

The results from this article suggest that AI can enhance programming education by providing instant feedback, customization, automated evaluation, and coding and debugging help. Above-mention can enhance student motivation and engagement, program performance, and access to education support. But the educational impact of AI is dependent upon how it is used. Over-reliance on solutions produced by the computer can reduce computational thinking, independent problem solving, creativity, and critical evaluation of code. Errors in or vulnerabilities of code, issues of plagiarism and academic honesty, privacy concerns, algorithmic bias, and unequal access are all concerns that have arisen with the use of AI that illustrate that AI cannot replace teachers, formal learning programs, or assessment methods.

Teachers plan activities that ask students to explain, adapt, test and challenge the code generated by AI instead of copying it from a source. Institutions should ensure approved tools, training of instructors, equal access, good academic-integrity policies, and effective data-protection procedures. Long-term and comparative research is needed to see the impact of various AI assistants on actual programming skills in educational contexts and across different levels. Policymakers could set up flexible governance structures that foster innovation and yet are transparent, easily accessible, accountable and learner safe.

Author Contributions: Author has contributed significantly to the development and completion of this article.

Funding: This article received no external funding.

Acknowledgments: The authors thank the Higher Institute of Technical Sciences, Misrata, Libya, for providing laboratory facilities and equipment.

Ethics approval: The study used public system-level data and involved no human participants.

Conflicts of Interest: The author(s) declare no conflict of interest.

References

[1] G. Lee, M. Yun, X. Zhai, and K. Crippen, "Artificial Intelligence in science education research: Current states and challenges," *J. Sci. Educ. Technol.*, vol. 35, no. 1, pp. 110–127, 2026.

[2] V. Chamola, D. Dave, I. Goyal, and S. Sharma, "Generative artificial intelligence in STEM education: A review of applications, benefits and challenges," *Expert Syst.*, vol. 43, no. 2, 2026.

[3] R. Che Ghazali, M. F. Abdul Hanid, M. N. H. Mohd Said, and H. Y. Lee, "The advancement of Artificial Intelligence in Education: Insights from a 1976–2024 bibliometric analysis," *J. Res. Technol. Educ.*, vol. 58, no. 4, pp. 906–922, 2026.

[4] I. Gidiotis and S. Hrastinski, "Imagining the future of artificial intelligence in education: a review of social science fiction," *Learn. Media Technol.*, vol. 51, no. 1, pp. 35–47, 2026.

[5] W. J. Kim, "Teachers' use of Generative Artificial Intelligence for designing science lessons in support of environmental science agency," *Res. Sci. Educ.*, vol. 56, no. 1, pp. 203–222, 2026.

[6] F. Khalifeh, R. Santiago, and R. Palau, "Redefining personalized learning in the artificial intelligence era: an updated systematic review from 2019 to 2025," *Smart Learn. Environ.*, vol. 13, no. 1, 2026.

[7] M. Alanazi, B. Soh, H. Samra, and A. Li, "The influence of artificial intelligence tools on learning outcomes in computer programming: A systematic review and meta-analysis," *Computers*, vol. 14, no. 5, p. 185, 2025.

[8] F. J. Agbo, C. Olivia, G. Oguibe, I. T. Sanusi, and G. Sani, "Computing education using generative artificial intelligence tools: A systematic literature review," *Comput. Educ. Open*, vol. 9, no. 100266, p. 100266, 2025.

[9] I. H. Y. Yim and J. Su, "Artificial intelligence literacy education in primary schools: a review," *Int. J. Technol. Des. Educ.*, vol. 35, no. 5, pp. 2175–2204, 2025.

[10] A. Rivera-Novoa and D. A. Duarte Arias, "Generative artificial intelligence and extended cognition in science learning contexts," *Sci. & Educ.*, vol. 35, no. 3, pp. 755–776, 2026.

[11] D. Pérez-Jorge, M. C. González-Afonso, A. G. Santos-Álvarez, Z. Plasencia-Carballo, and C. de L. Á. Perdomo-López, "The impact of AI-driven Application Programming Interfaces (APIs) on educational information management," *Information (Basel)*, vol. 16, no. 7, p. 540, 2025.

[12] G. Fan, D. Liu, R. Zhang, and L. Pan, "The impact of AI-assisted pair programming on student motivation, programming anxiety, collaborative learning, and programming performance: a comparative study with traditional pair programming and individual approaches," *Int. J. STEM Educ.*, vol. 12, no. 1, 2025.

[13] H. Güner and E. Er, "AI in the classroom: Exploring students' interaction with ChatGPT in programming learning," *Educ. Inf. Technol. (Dordr.)*, vol. 30, no. 9, pp. 12681–12707, 2025.

[14] C. Matobobo, P. D. N. Ncube, N. L. Ngesimani, G. P. Dzvapatsva, and E. Chinhamo, "Enhancing computational thinking and problemsolving in programming education through generative AI: A scoped review," in *2025 IEEE Global Engineering Education Conference (EDUCON)*, 2025, pp. 1–9.

[15] A. Park and T. Kim, "Code suggestions and explanations in programming learning: Use of ChatGPT and performance," *Int. J. Manag. Educ.*, vol. 23, no. 2, p. 101119, 2025.

[16] D. Kohen-Vacs, M. Usher, and M. Jansen, "Integrating generative AI into programming education: Student perceptions and the challenge of correcting AI errors," *Int. J. Artif. Intell. Educ.*, vol. 35, no. 5, pp. 3166–3184, 2025.

[17] S. Elnaffar, F. Rashidi, and A. Z. Abualkishik, "Teaching with AI: A systematic review of chatbots, generative tools, and tutoring systems in programming education," *Int. J. Learn. Teach. Educ. Res.*, vol. 25, no. 1, pp. 1–28, 2026.

[18] J. Katona and K. I. K. Gyonyoru, "AI-based adaptive programming education for socially disadvantaged students: Bridging the digital divide," *TechTrends*, vol. 69, no. 5, pp. 925–942, 2025.

[19] S. Boguslawski, R. Deer, and M. G. Dawson, "Programming education and learner motivation in the age of generative AI: student and educator perspectives," *Inf. Learn. Sci.*, vol. 126, no. 1/2, pp. 91–109, 2025.

[20] A. Garg, K. Nisumba Soodhani, and R. Rajendran, "Enhancing data analysis and programming skills through structured prompt training: The impact of generative AI in engineering education," *Computers and Education: Artificial Intelligence*, vol. 8, no. 100380, p. 100380, 2025.

[21] P. J. Aguilar-Cruz and S. Z. Salas-Pilco, "Teachers' perceptions of artificial intelligence in Colombia: AI technological access, AI teacher professional development and AI ethical awareness," *Technol. Pedagog. Educ.*, vol. 34, no. 2, pp. 219–238, 2025.

[22] M. Sivasakthi and A. Meenakshi, "Generative AI in programming education: Evaluating ChatGPT's effect on computational thinking," *SN Comput. Sci.*, vol. 6, no. 5, 2025.

[23] T.-C. Yang, Y.-C. Hsu, and J.-Y. Wu, "The effectiveness of ChatGPT in assisting high school students in programming learning: evidence from a quasi-experimental research," *Interact. Learn. Environ.*, vol. 33, no. 6, pp. 3726–3743, 2025.

[24] T.-C. Hsu and M.-S. Chen, "Effects of students using different learning approaches for learning computational thinking and AI applications," *Educ. Inf. Technol. (Dordr.)*, vol. 30, no. 6, pp. 7549–7571, 2025.

[25] Z. Yang and J. Shin, "The impact of Gen AI on art and design program education," *Des. J.*, vol. 28, no. 2, pp. 310–326, 2025.

[26] S. S. Oyelere and K. Aruleba, "A comparative study of student perceptions on generative AI in programming education across Sub-Saharan Africa," *Comput. Educ. Open*, vol. 8, no. 100245, p. 100245, 2025.

[27] J. Ma *et al.*, "The role of artificial intelligence in shaping nursing education: A comprehensive systematic review," *Nurse Educ. Pract.*, vol. 84, no. 104345, p. 104345, 2025.

[28] P. M. Simelane and J. Kittur, "Use of generative artificial intelligence in teaching and learning: Engineering instructors' perspectives," *Comput. Appl. Eng. Educ.*, vol. 33, no. 1, 2025.

[29] J. Hong and K. Kim, "Impact of AIoT education program on digital and AI literacy of elementary school students," *Educ. Inf. Technol. (Dordr.)*, vol. 30, no. 1, pp. 107–130, 2025.

[30] J. Liang, J. M. Stephens, and G. T. L. Brown, "A systematic review of the early impact of artificial intelligence on higher education curriculum, instruction, and assessment," *Front. Educ.*, vol. 10, no. 1522841, 2025.

[31] M. M. Msambwa, Z. Wen, and K. Daniel, "The impact of AI on the personal and collaborative learning environments in higher education," *Eur. J. Educ.*, vol. 60, no. 1, 2025.

[32] L. V. Mulaudzi and J. Hamilton, "Lecturer's perspective on the role of AI in personalized learning: Benefits, challenges, and ethical considerations in higher education," *J. Acad. Ethics*, vol. 23, no. 4, pp. 1571–1591, 2025.

[33] A. Mouta, E. M. Torrecilla-Sánchez, and A. M. Pinto-Llorente, "Comprehensive professional learning for teacher agency in addressing ethical challenges of AIED: Insights from educational design research," *Educ. Inf. Technol. (Dordr.)*, 2024.

[34] J. Garzón, E. Patiño, and C. Marulanda, "Systematic review of artificial intelligence in education: Trends, benefits, and challenges," *Multimodal Technol. Interact.*, vol. 9, no. 8, p. 84, 2025.

[35] M. M. Alnsour, L. Qouzah, S. Aljamani, R. A. Alamoush, and M. K. AL-Omiri, "AI in education: enhancing learning potential and addressing ethical considerations among academic staff—a cross-sectional study at the University of Jordan," *Int. J. Educ. Integr.*, vol. 21, no. 1, 2025.

[36] E. Elbasi, M. Nadeem, Y. I. Alzoubi, A. E. Topcu, and G. Varghese, "Machine learning in education: Innovations, impacts, and ethical considerations," *IEEE Access*, vol. 13, pp. 128741–128770, 2025.

[37] S. Khan *et al.*, "Harnessing AI for sustainable higher education: ethical considerations, operational efficiency, and future directions," *Discov. Sustain.*, vol. 6, no. 1, 2025.

[38] M. Lepp and J. Kaimre, "Does generative AI help in learning programming: Students' perceptions, reported use and relation to performance," *Comput. Hum. Behav. Rep.*, vol. 18, no. 100642, p. 100642, 2025.