African Journal of Advanced Pure and Applied Sciences (AJAPAS)
Online ISSN: 2957-644X
Volume 5, Issue 2, 2026
Page No: 182-191
Website: https://aaasjournals.com/index.php/ajapas/index
ISI 2025: 1.126 **SJIFactor 2024: 6.752** **معامل التأثير العربي: 1.62**

Using Artificial Intelligence in Programming Language Education: A Review of Tools, Applications, Benefits, and Challenges

Ashraf Faraj Saed Albarki [1], Esam Miftah Abdulnabi Aboudoumat [*2]
[1] Computer Department, College of Arts and Sciences - Qumins, University of Benghazi, Libya
[2] Computer Department, College of Science and Technology, Qumins, Benghazi, Libya

استخدام الذكاء الاصطناعي في تعليم لغات البرمجة: مراجعة للأدوات والتطبيقات والفوائد والتحديات

أشرف فرج سعد البركي [1]، عصام مفتاح عبد النبي ابودومات [*2]
[1] قسم علوم الحاسوب، كلية الآداب والعلوم، قمينس، جامعة بنغازي، ليبيا
[2] قسم علوم الحاسوب، كلية العلوم والتقنية، قمينس، بنغازي، ليبيا

[*]Corresponding author: esam.mouftah@gmail.com

Received: March 07, 2026	Accepted: May 16, 2026	Published: May 25, 2026

Abstract:
In recent years, there has been significant development in the use of artificial intelligence (AI) applications and tools in the educational field, particularly in learning programming and programming languages This review paper examines the role of AI in programming language education by analyzing major tools, educational applications, advantages, and associated challenges. The paper also discusses the most important AI-based tools used in programming education, such as Deep Code and GitHub Copilot, and analyzes the advantages and disadvantages of these tools and their impact on students and teachers. The results indicate that using AI in programming education contributes to improving programming comprehension, reducing errors, supporting self-learning, and accelerating the software development process.

Keywords: Artificial Intelligence, Programming Learning, Programming Languages, Smart Learning.

الملخص
شهدت السنوات الأخيرة تطوراً ملحوظاً في استخدام تطبيقات وأدوات الذكاء الاصطناعي في المجال التعليمي، لا سيما في تعلم البرمجة ولغاتها. تتناول هذه الورقة البحثية دور الذكاء الاصطناعي في تعليم لغات البرمجة من خلال تحليل الأدوات الرئيسية، والتطبيقات التعليمية، والمزايا، والتحديات المرتبطة بها. كما تناقش الورقة أهم أدوات الذكاء الاصطناعي المستخدمة في تعليم البرمجة، مثل Deep Code وGitHub Copilot، وتحلل مزايا وعيوب هذه الأدوات وتأثيرها على الطلاب والمعلمين. تشير النتائج إلى أن استخدام الذكاء الاصطناعي في تعليم البرمجة يُسهم في تحسين فهم البرمجة، وتقليل الأخطاء، ودعم التعلم الذاتي، وتسريع عملية تطوير البرمجيات.

الكلمات المفتاحية: الذكاء الاصطناعي، تعلم البرمجة، لغات البرمجة، التعلم الذكي.

1. Introduction

The use of artificial intelligence in contemporary education is a strategic driver of change in the educational process. Artificial intelligence represents an integrated knowledge system that transcends the concept of traditional technology. It offers intelligent, interactive applications capable of personalizing learning paths, enhancing cognitive interaction, and improving the quality of educational outcomes in an unprecedented way [1].Artificial intelligence (AI) is defined as a field concerned with developing computer systems that mimic human

cognitive abilities. These applications are capable of performing advanced cognitive functions such as making informed decisions, understanding and analyzing natural languages, and learning from experiences and available data. Waleed and Yaqoub Al-Shaqsi (2025) define AI operationally as:
A set of advanced computer applications and systems that can simulate aspects of human intelligence, enabling them to accomplish complex tasks such as pattern recognition, decision-making, problem-solving, and the ability to learn and adapt to different variables. In the field of education specifically,
the term refers to the use of algorithms and intelligent systems to develop the educational process, enrich students' learning experience, and enhance teachers' teaching methods and practices [1].
Artificial intelligence (AI) is a key pillar of computer science, experiencing rapid development that has led to the emergence of numerous software programs and applications that have permeated various aspects of daily life, with expectations of continued expansion in the coming years. Key AI applications include expert systems, natural language processing, speech recognition, pattern recognition, machine programming, and robotics. Several of these applications have been integrated into the educational field through what are known as intelligent learning systems and educational expert systems, contributing to the development of teaching and learning processes and improving their outcomes. These applications also encompass knowledge-based systems and AI-powered educational programs that rely on logic and symbolic rules to deliver educational content, guide learners, and evaluate their performance, closely mimicking the role of a human teacher and enhancing the effectiveness of the educational process.
AI-based educational programs aim to meet the learning needs of diverse student groups and achieve multiple educational objectives by providing flexible learning environments that support interaction and communication among learners, facilitate access to digital resources, and promote the principle of decentralized learning. These programs also contribute to integrating students into the educational process in diverse and meaningful ways, supporting more effective learning. In light of these rapid developments, artificial intelligence (AI) is expected to play a pivotal role in developing the educational process and improving the quality of its outcomes.
AI and education can be viewed as two complementary elements within a single system. Education aims to develop students' knowledge and skills, while AI contributes to enhancing understanding of the mechanisms of thinking, knowledge, and intelligent behavior. Interactive learning environments based on AI technologies have been developed, providing opportunities for direct interaction between students, computers, and smart devices, helping them discover new concepts more effectively. The results of using these environments have shown positive effects on several variables related to the learning process, in addition to developing various thinking and problem-solving skills [2].
AI contributes to enhancing the experience of programmers in multiple fields by simplifying programming processes and providing advanced analytical tools, which positively impacts increased productivity and improved code quality. Artificial intelligence also enables the provision of effective solutions to technical challenges, and supports the software development process by accelerating code writing, detecting errors, and suggesting appropriate improvements [3].

Research Problem Statement:
Many students, especially beginners, face numerous problems and challenges while learning programming. Among the most significant are:
• Difficulty understanding fundamental programming concepts.
• Rapid transitions between programming concepts can cause confusion for beginners.
• Frequent programming errors and the inability to detect, understand, and correct them.
• Lack of immediate support during learning and difficulty organizing code, leading to disorganized code.
• Varying levels of understanding and comprehension among students in the same class.
• Difficulty in manually evaluating and tracking code.
All these problems have led to the widespread use of artificial intelligence (AI) technologies and tools to enhance the programming learning process and provide a more efficient and interactive learning experience.

Research Objectives:
This research aims to:
This review paper aims to:
• Examine the role of artificial intelligence in programming language education.
• Identify the major AI-based tools used in programming learning.
• Analyze the main educational benefits of AI in programming instruction.
• Discuss the pedagogical, ethical, and technical challenges associated with AI use.
• Highlight future directions for more effective integration of AI in programming education.
•Compare traditional education with AI-based education.

2.Review Methodology

This research paper adopts a narrative review methodology to analyze the use of artificial intelligence in programming language education. To enhance the review, priority was given to articles published in peer-reviewed journals and conference papers, mainly published between 2020 and 2025. The literature was defined through a thematic search focusing on terms such as "artificial intelligence in programming education," "AI programming assistants," "automated programming feedback," "adaptive programming learning," , Because the goal of this paper is to provide an organized academic synthesis rather than a statistical meta-analysis, the review emphasizes thematic comparison, convergence across studies, and reported gaps in current research.

3. Artificial Intelligence in Programming Education:

Artificial intelligence is used in programming education in several ways, including:

1.3 AI Coding Tools:

AI-powered coding tools provide automatic source code suggestions as you write, enabling students and programmers to automate code, suggest appropriate algorithms, debug code, and clarify code functions. The development of these tools relies on machine learning and artificial intelligence technologies, which contribute to simplifying and accelerating programming processes, reducing human error, and improving the efficiency and quality of software development.

The development of intelligent coding tools is a vital field that significantly enhances developer productivity and improves the programming experience. The design of these tools is based on modern technologies such as machine learning and artificial intelligence, aiming to provide more interactive and effective programming environments. Among these tools, intelligent source code analysis tools are essential, as advanced analysis techniques enable error identification, application performance optimization, and support for maintenance and continuous development. Smart editing tools also integrate real-time guidance and code hints while writing, increasing writing efficiency and reducing coding errors.

Furthermore, smart software project management tools enable the organization of software teams, tracking of project progress, and efficient workflow optimization. Some smart tools even support automatic code generators and self-compilation, contributing to faster development and reduced manual effort. Investing in the development of these smart software tools reflects a qualitative shift in software development methodology and contributes to building applications with higher performance, efficiency, and improved sustainability [3].

Recent years have witnessed the emergence of several software tools and platforms that rely on artificial intelligence technologies to facilitate software development and improve developer productivity. Among the most prominent tools that have revolutionized the world of programming are:

1. **GitHub Copilot**: This is an intelligent programming assistant developed by Microsoft that uses the GPT code generation model. It is distinguished by its ability to understand the project context and suggest complete code snippets that are compatible with the code style used, thus contributing to faster programming and improved code quality. 2. Tabnine: This is one of the first tools to use artificial intelligence to automatically complete code. It learns from thousands of software projects to predict the code a developer needs before writing it completely, thus enhancing development speed and efficiency.
3. **Deep Code**: This tool relies on machine learning algorithms to analyze code and detect errors and security vulnerabilities. It also provides intelligent reports to developers showing how to improve the code to become more secure and efficient.
4. **Kite:** This tool offers intelligent suggestions while writing code in more than 16 programming languages, based on the analysis of open-source code projects to provide the most common and accurate solutions.
5. **Amazon Code Whisperer**: An AI platform developed by Amazon to provide real-time recommendations to programmers while writing code within the AWS environment, reducing the time required to develop cloud applications and enhancing work efficiency. [4]

3.2 Automatic evaluation systems:

Error analysis is a crucial element of the software development lifecycle, given the negative impact errors can have on system efficiency and functionality. AI-powered software tools have significantly enhanced error detection by performing automated source code analysis with greater accuracy and speed. These tools rely on recognizing programming patterns and identifying recurring errors, then provide immediate recommendations to help developers address problems more effectively.

Furthermore, some of these tools offer detailed analytical reports that include descriptions of errors and their potential causes. This enables developers to pinpoint the root cause of the problem rather than simply addressing its symptoms. This approach improves developer learning by enhancing their ability to detect and avoid errors in the future while writing code. Relying on such analyses also leads to improved code quality and reduced time and effort spent on debugging and maintenance. Consequently, automated code evaluation simplifies debugging tasks

that typically require considerable time and effort from programmers, positively impacting development efficiency and the quality of software output [5].

3.3 Adaptive Learning

The concept of adaptive learning emerged as a solution to a common problem in traditional educational systems, which often provide the same educational content and resources to all students, regardless of their different learning styles and cognitive abilities. Adaptive learning aims to tailor educational content and learning paths to students' abilities and cognitive levels, thus reducing cognitive burdens and enhancing the efficiency of the educational process [2].

Adaptive learning is one of the most prominent practical applications of artificial intelligence in education, as it allows for the personalization of educational resources and activities to meet the individual needs of each learner. This process relies on analyzing student performance data, where AI algorithms adjust the difficulty and complexity of the educational content according to the student's strengths and weaknesses, with the goal of improving learning effectiveness and focusing on areas that require further development.

4. Using Artificial Intelligence with Programming Languages:

Artificial intelligence provides programmers with intelligent analytics and guidance during the programming process, enabling them to make informed decisions and improve software development performance. These analytics are an essential part of technical operations and strategic decision-making, providing in-depth data understanding and helping to uncover vital trends and patterns within software projects. They leverage machine learning techniques and intelligent algorithms to analyze data with high efficiency and deliver valuable insights to both programmers and organizations [3].

Modern tools and technologies also enable the provision of intelligent, customized guidance tailored to individual or organizational needs, relying on advanced analytics to support strategic decision-making. One of the most significant advantages of these systems is their ability to continuously adapt to environmental changes and external variables, with the potential to enhance and update analytical models to ensure ongoing accuracy and effectiveness. Therefore, the use of analytics and intelligent guidance strengthens the ability to make informed decisions across various fields, including software development, business, marketing, and big data management.

The practical impact of artificial intelligence is clearly demonstrated through tools like GitHub Copilot and Tabnine, which utilize deep learning techniques for programming languages to automatically generate code based on simple commands written by the developer. These tools not only accelerate the programming process but also reduce grammatical and logical errors through continuous learning from millions of lines of open-source code and by suggesting optimal solutions to the problems presented [4].

5. Comparison between Traditional Education and Education Using Artificial Intelligence:

The use of artificial intelligence in education does not aim to replace the teacher or eliminate the role of traditional education. Rather, it is used as a supportive tool that contributes to improving the quality and efficiency of the educational process. Although AI technologies can deliver customized educational content at high speed and provide immediate feedback to learners, the teacher remains central to the educational process due to their ability to provide educational guidance, psychological support, and develop students' creative and critical thinking skills—human aspects that intelligent systems cannot fully replicate.

Successful future education is based on the integration of the human element of traditional education with the capabilities of modern technologies. It combines the teacher's role in guidance and human interaction with the capabilities of artificial intelligence in personalization, analysis, and the flexible delivery of content, thus contributing to improved learning quality and more effective educational outcomes. The following is a comparison between traditional education and education using artificial intelligence[6]:

Table 1: comparison between traditional education and education using artificial intelligence.

Standard	Traditional Education	Education Using Artificial Intelligence
Learning Method	Teacher-led learning	Interactive learning based on smart distance learning systems
Learning Speed	Constant for all students	Different according to each learner's level
Interaction (Learning Environment)	Limited to the lesson period only	Continuous interaction with the system through online virtual classrooms
Customization	Same content for all students	Content tailored to the student's level

Standard	Traditional Education	Education Using Artificial Intelligence
Programming Debugging	Depending on the teacher	Instant correction and solution suggestions
Feedback	Often delayed	Immediate and real-time
Hands-on Training	Limited exercises	Smart exercises that adapt automatically
Self-Learning	Limited	Highly efficient and flexible
Accessibility (Flexibility)	Time and place-bound	Available anytime, anywhere
Performance Monitoring	Traditional through tests	Intelligent student assessment
Learning Materials	Relies on books and tangible learning resources	Utilizes multimedia materials such as videos, interactive modules, simulations, and e-books

6. Advantages of Using Artificial Intelligence in Programming Education:

The advantages of artificial intelligence in programming education include:

• Code Analysis:

AI tools contribute to code analysis by detecting grammatical and logical errors and providing appropriate suggestions for correction and improvement. These tools also provide detailed explanations of the functions of different code components, helping students understand the structure of complex programs and grasp how they work. This, in turn, enhances their skills in writing more accurate and efficient code.

• Accelerating the Development Process:

The ability to automatically generate code snippets based on user requirements helps reduce the time and effort spent writing repetitive code. It also allows developers and students to focus on designing algorithms and solving problems instead of getting bogged down in implementation details, thus improving the efficiency of the learning and development process.

• Improving Quality and Reducing Errors:

The ability to detect problems in the early stages of the development process reduces their recurrence later on. These tools also provide immediate correction of programming errors and offer suggestions for improvement, helping learners quickly understand their mistakes and contributing to faster learning and more efficient coding.

• Reduced Costs:

By reducing expenses on training, educational materials, time, and human resources, learning programming becomes more accessible and available to a wider range of learners.

• Increased Developer Productivity:

AI tools contribute to fostering creativity and innovation by reducing the routine tasks associated with writing code. These tools help generate initial code snippets and suggest solutions to programming problems, allowing students to focus on understanding fundamental concepts and developing analytical thinking skills, rather than getting bogged down in implementation details. This contributes to improving the quality and efficiency of learning.

• Enhanced Security: Continuous code analysis reduces security vulnerabilities.[4]

• Commentary:

AI can read code and write comments explaining the function of each line or function, making the program easier to understand.

• Enhanced Knowledge and Essential Skills for Understanding Technology • Motivating learning by receiving feedback when encountering difficulties boosts their self-confidence[7] [4].

7. Challenges and Drawbacks:

The most prominent drawbacks of using artificial intelligence tools in programming education include:

• Misuse:

The ease of access to and integration of these tools may tempt students to shorten the learning process instead of delving deeply into the subject matter. This weakens the learning experience, as students become passive recipients rather than active participants in the problem-solving process.

• Reduces students' critical thinking abilities:

It increases their dependence on technology instead of teaching them how to complete tasks independently.

• Learners may rely on generating code without a true understanding of programming logic, which weakens analytical thinking skills.

• Over-reliance on AI tools: This may reduce programmers' skills over time. This can lead to a superficial and limited understanding of fundamental programming concepts and a decline in their ability to solve problems independently.

• Obtaining solutions directly reduces students' attempts at self-reflection and discovering solutions independently.
• Privacy and data security issues: Because AI tools sometimes need access to code and projects.
Bias in algorithms: Proposed code may result from incomplete training data.
Difficulty in adoption in traditional institutions: Due to resistance to change or a lack of technical expertise [7][4].

8.Related Work:

Samah Atiq Salem Al-Mousa (2024) study, "The Role of Artificial Intelligence in Improving Programmers' Experience in Various Fields," examined the impact of AI technologies on developing the programmers' work environment and enhancing their efficiency. The study aimed to analyze the challenges facing programmers and explore how AI tools can be employed to simplify programming processes and accelerate software development. The study indicated that AI contributes to improved productivity through big data analysis, the provision of intelligent software tools, and enhanced interaction between programmers and systems. It also provides intelligent analytics and guidance that support decision-making during the programming process. The results showed that using AI technologies helps in developing advanced tools, reducing programming errors, and improving the integration of different technologies, leading to an enhanced programmer experience and increased software development efficiency. The study recommended training programmers on the use of intelligent tools, supporting research and development in the field of integrating AI into programming, and encouraging the adoption of these technologies to improve programming performance[3].

The study by Waleed bin Zaher and Yaqoub bin Zaher Al-Shaqsi (2025), titled "The Impact of Using Artificial Intelligence Applications in Education on Academic Achievement, Student and Teacher Satisfaction, and Self-Efficacy," aimed to identify the effect of employing artificial intelligence applications in the educational process on academic achievement, student and teacher satisfaction levels, and teacher self-efficacy. The study adopted a quasi-experimental design, applying it to a sample of 600 tenth-grade students and 16 teachers, divided into two groups: an experimental group that used artificial intelligence applications and a control group that relied on traditional methods. The results showed statistically significant differences favoring the experimental group, with a 15% improvement in academic achievement, a 25% increase in student engagement and satisfaction with the educational process, and a rise in teacher satisfaction and self-efficacy compared to the control group. The study recommended the adoption of artificial intelligence applications in education, the provision of appropriate infrastructure, and the training of teachers and students to ensure the effective use of these technologies and improve learning outcomes[1].

Rina Zviel-Girshin (2024) study, "Advantages and Disadvantages of AI Tools in Programming Education for Beginners," examined the impact of AI tools in introductory programming courses. It aimed to analyze how beginners use these tools and identify their advantages and challenges in developing programming skills. The study employed a mixed-methods approach, collecting data from 73 engineering student teams over a 12-week period. Questionnaires and usage reports were used to analyze engagement and satisfaction levels. The results showed a significant increase in students' knowledge and use of AI tools, primarily for writing code comments (91.7%), detecting and correcting errors (80.2%), and searching for information (68.5%). The study also indicated that these tools help accelerate learning, provide immediate support, and reduce the cognitive load for beginners, thus improving the programming learning experience. Conversely, the results pointed to several challenges, including over-reliance on the tools, the potential for academic dishonesty, a weak grasp of fundamental concepts, and the possibility of producing inaccurate or misleading code. The study concluded that artificial intelligence tools should be used in a balanced way with traditional methods to ensure the development of critical thinking and problem-solving skills among beginners[7].

A study by M. Messer, N. Brown, and M. Kölling. (2023) conducted a systematic review of automated assessment and feedback tools in programming education, analyzing 121 recent studies. The results showed that most of these tools focus on evaluating solution validity using techniques such as unit testing and static analysis, with relative neglect of code quality aspects like maintainability and readability. The study also revealed that while most systems rely on fully automated assessment to provide immediate feedback, this feedback is often limited. Although some machine learning applications have emerged, their use remains restricted due to the need for large training datasets. The study further highlighted research gaps related to improving feedback quality and expanding the scope of assessment to encompass deeper programming skills[8].

The study by Katona & Gyonyoru (2025) aimed to investigate the impact of AI-based adaptive programming instruction on enhancing learning outcomes and integration levels among students facing challenging social circumstances, with the goal of bridging the digital divide and promoting social mobility. The study employed a blended learning approach, applying it to a sample of 122 students from four Hungarian universities. These students were classified as socially disadvantaged based on the Social Conditions Index (SCI). They were divided into an experimental group that studied in an adaptive environment supported by ChatGPT, and a control group that studied using a standardized, traditional curriculum over a period of 13 weeks. The results showed a

significant advantage for the experimental group in programming knowledge acquisition and in all dimensions of integration (behavioral, emotional, and cognitive) compared to the control group, with the experimental group registering a large effect size of 1.40. Statistical analysis (ANCOVA) also confirmed that the artificial intelligence system had a tangible positive impact on student outcomes regardless of their economic and social backgrounds, indicating the ability of adaptive learning environments to reduce educational gaps and provide equal opportunities for the most disadvantaged students[9].

The study (F. Ali, A. Ahmed, M. A. Alipour, and H. Terashima-Marin., 2025) aimed to explore the factors influencing computer science students' acceptance and adoption of AI-based programming assistants (AI-CAs), with a particular focus on the role of trust and learning motivation. The study applied the Technology Acceptance Model (TAM), expanded to include six external factors: interest in learning, achievement goals, self-criteria (within learning motivation), trust, skepticism, and insecurity (within trust factors). The researchers employed a quantitative approach, analyzing the responses of 311 university students in Mexico using structural equation modeling (SEM). The results showed that "perceived usefulness" is the strongest predictor of students' intention to use these tools. Interest in learning, trust, and achievement goals significantly enhance students' perception of the usefulness and ease of use of AI-CAs. Conversely, the study revealed a limited role for negative emotional factors such as skepticism and insecurity in influencing the adoption decision, suggesting that students focus more on the cognitive and practical aspects (ease of use and usefulness) when dealing with AI technologies in the context of programming education[10].

The study (P. Manorat, S. Tuarob, and S. Pongpaichet., 2025) aimed to provide a comprehensive and in-depth understanding of the role of artificial intelligence (AI) and machine learning (ML) in computer programming education, going beyond previous studies that focused on limited aspects. The researchers analyzed 119 research papers published between 2012 and 2023 to categorize AI applications across the entire pedagogical (teaching) process. The study's findings revealed a four-part classification of technical intervention areas, including: student assessment and correction of programming tasks, generation of customized automated feedback, plagiarism detection, and predicting student performance and identifying those at risk of academic failure. The review also highlighted a significant advancement in the use of Large Language Models (LLMs) and deep learning to develop intelligent tools that can assist teachers in designing educational content and providing self-directed learning environments for students, thereby enhancing the efficiency of the educational process. The study concluded by identifying future research gaps, emphasizing the need to balance the technical benefits of artificial intelligence with ethical and educational considerations in programming learning environments[11].

The study (J. Nathaniel, S. S. Oyelere, J. Suhonen, and M. Tedre 2025) conducted a systematic review of 40 pilot studies to assess the effectiveness of integrating generative artificial intelligence (GenAI) tools such as ChatGPT and GitHub Copilot into programming education and their impact on students' foundational skills and higher-order thinking. The review revealed that the successful integration of these technologies depends fundamentally on adopting intentional teaching strategies, designing innovative assessments, and structuring integration processes to go beyond simply providing the tool and to align with the curriculum. The study also highlighted critical challenges, most notably the risk of students becoming overly reliant on smart tools, which could weaken their programming logic, as well as technical constraints related to limited accessibility features and algorithmic bias. The study concluded by proposing a framework that balances investing in technological innovation with maintaining the quality of educational outcomes, emphasizing the need to redefine educational roles to ensure the development of critical thinking among programming students in the era of generative AI[12].

The study by(Y. Xue, H. Chen, G. R. Bai, R. Tairas, and Y. Huang,2024) aimed to investigate the role of the ChatGPT model in foundational programming courses (CS1) through a controlled experiment involving 56 participants. The study assessed the impact of using generative artificial intelligence on learning outcomes and student behavior during programming assignments. The methodology involved dividing participants into an experimental group, which was allowed to use ChatGPT 3.5 and internet resources, and a control group, which was allowed to use all resources except artificial intelligence. Both groups were tasked with designing and programming UML diagrams. The results revealed that the experimental group significantly outperformed the control group in the quality of completed tasks (UML design and programming) and in speed of completion. However, the post-evaluation results did not show any significant differences in deep learning achievement between the two groups. The study highlighted, through an analysis of screen recordings, that students tend to use artificial intelligence as a tool for generating code and correcting errors, which enhances productivity. However, it raises questions about the extent to which students independently grasp basic programming concepts, which calls for a reconsideration of the assessment methods used in light of the availability of these technologies[13].

The study by(Y. Jing, H. Wang, X. Chen, and C. Wang,2024) investigated the factors influencing learners' effective use of the ChatGPT model for solving programming problems, focusing on the Python programming language as an applied model. The methodology was based on measuring learner performance and correlating it with specific variables, including AI literacy, prior programming knowledge, familiarity with the tool, and intent to use it. The results showed that AI literacy—particularly awareness of its mechanisms and how to use it—is one of the most important factors determining a student's success in employing the technology to solve problems. The

study also demonstrated that possessing a solid programming language foundation significantly improves the efficiency of interaction with ChatGPT, while prior intent to use it did not have a major impact on actual performance, although it showed a noticeable improvement after the experiment. The study concluded by offering suggestions for enabling AI-generated content learning (AIGC), emphasizing the need to enhance students' interaction and critical analysis skills to ensure they can make the most of these tools in a programming context[14].

Taken together, these studies suggest that AI can improve programming education when it is integrated carefully, supported pedagogically, and balanced with human guidance.

9.Key Findings from the Literature:

Based on the reviewed literature, AI tools can significantly improve the speed and efficiency of programming learning, automated feedback helps reduce coding errors, intelligent assistants increase learner productivity, and adaptive systems are useful for addressing learner differences. Recent review and empirical studies also suggest that the educational impact of AI depends strongly on instructional design, assessment structure, learner AI literacy, and the extent to which these tools are used for guided support rather than answer substitution [8], [11]–[14].

At the same time, the reviewed studies emphasize that the educational value of AI is strongest when these tools are used to support learning rather than replace reasoning or instruction. The major risks remain over-reliance, reduced critical thinking, privacy concerns, inaccurate suggestions, and academic misconduct.

Through studying the topic, the researcher concluded that using artificial intelligence in programming education leads to:

- Improved speed of programming learning
- Reduced programming errors.
- Increased interaction between the student and the system.
- Improved code quality.
- Support for self-directed learning.

10. Conclusion:

This review paper examined the role of artificial intelligence in programming language education, focusing on tools, applications, benefits, and challenges. The reviewed literature indicates that AI has become an important support technology in programming education through intelligent coding assistants, automated code evaluation systems, and adaptive learning environments.

The main benefits highlighted in the literature include faster feedback, improved error detection, support for self-learning, better code quality, and increased learner engagement. At the same time, the literature points to significant challenges, including over-reliance, reduced critical thinking, academic integrity concerns, privacy issues, and the possibility of misleading AI-generated outputs.

Overall, the findings suggest that AI can make programming education more effective, interactive, and flexible when used within a carefully designed pedagogical framework. The most appropriate educational model is not one that replaces teachers with intelligent systems, but one that combines human teaching expertise with the analytical and adaptive strengths of artificial intelligence.

Table 2. Representative studies reviewed in this paper

Study	Year	Focus	Main Finding	Limitation
Al-Shaqsi & Al-Shaqsi	2025	AI applications in education	Improved achievement, engagement, and satisfaction	General education context, not programming-specific
Al-Mousa	2024	AI and programmer experience	Higher productivity and stronger intelligent support tools	Focuses on programmers broadly rather than formal classrooms
Zviel-Girshin	2024	Novice programming education	Faster support, error help, and reduced cognitive load	Highlights over-reliance and academic dishonesty risks
Messer et al.	2023	Automated grading and feedback	Near-instant feedback and support for repeated resubmissions	Most tools still focus mainly on correctness
Katona & Gyonyoru	2025	Adaptive programming education	Higher achievement and engagement in an AI-adaptive environment	Focused on a specific disadvantaged learner context
Ali et al.	2025	Adoption of AI coding assistants	Usefulness, trust, and motivation strongly influence adoption	Addresses adoption more than direct learning gains

Manorat et al.	2025	Systematic review of AI in programming education	Field shifted toward classroom implementation, assessment, and pedagogical support	Broad review; not limited to one educational level or single tool
Nathaniel et al.	2025	GenAI integration in programming education	Effective integration depends on curriculum alignment, assessment design, and structured support	Focused on empirical GenAI studies and integration quality
Xue et al.	2024	ChatGPT use in CS1	AI support can help with programming tasks but raises dependence and learning-quality concerns	Introductory course context only
Jing et al.	2024	ChatGPT for solving programming problems	AI literacy and prior knowledge strongly affect effective educational use	Focused on a quasi-experimental problem-solving setting

11.Future Work:

This field can be further developed in the future through:

• Developing fully intelligent programming education systems (IDEs) and integrating artificial intelligence with development environments.

• Creating interactive programming learning platforms, supporting additional programming languages, and developing more accurate assessment systems.

• improving automated feedback so that it explains errors rather than only correcting them.

12.Recommendations:

• It is recommended to integrate artificial intelligence tools, technologies, and their educational applications into programming courses within computer science and software engineering programs at universities. This will contribute to enhancing the efficiency of the educational process and aligning it with modern technological advancements.

• It is also recommended to organize regular training courses for beginners in programming. These courses should aim to introduce them to artificial intelligence concepts and their applications in supporting the teaching and learning processes, particularly in the context of learning programming languages, thereby enhancing their academic and skill-based readiness.

Compliance with ethical standards

Disclosure of conflict of interest

The authors declare that they have no conflict of interest.

References

[1] Waleed bin Zaher bin Sulaiman Al-Shaqsi, Yaqoub bin Zaher bin Sulaiman Al-Shaqsi, “The role of artificial intelligence applications in education on academic achievement, student and teacher satisfaction, and self-efficac ”, Journal of Human Science, ISSN: 3079-9384, DOI: https://doi.org/10.63496/ejhs.Vol1.Iss3.82.

[2] Lina Ahmed Khalil Al-Farani,Hania Abdul-Razzaq Ahmed Fatani," From Adaptation to Accreditation: Artificial Intelligence Applications in Intermediate Schools ", The Comprehensive Multidisciplinary Electronic Journal for Publishing Scientific and Educational Research (MECSJ),Issue 21, January 2020,ISSN: 2617-9563.

[3] Samah Atiq Salem Al-Mousa," The Role of Artificial Intelligence in Enhancing the Programmer Experience in Various Fields ", Arab Society Journal for Publishing Scientific Studies -Issue No. 39 - 2024-2-12,ISSN: 2958-6798.

[4] [Online] Available:.https://nahil.com.sa/ar/%D8%A7%D9%84%D8%B0%D9%83%D8%A7%D8%A1-%D8%A7%D9%84%D8%A7%D8%B5%D8%B7%D9%86%D8%A7%D8%B9%D9%8A-%D9%81%D9%8A-%D8%AA%D8%B7%D9%88%D9%8A%D8%B1-%D8%A7%D9%84%D8%A8%D8%B1%D9%85%D8%AC%D9%8A%D8%A7%D8%AA-%D8%A3/.[Accessed: 01-04-2026, Time: 12:00 AM].

[5][Online]Available:https://lezrweb.com/2026/02/13/%D9%85%D8%B3%D8%A7%D8%B9%D8%AF%D8%A7%D8%AA-%D8%A8%D8%B1%D9%85%D8%AC%D9%8A%D8%A9-%D8%B0%D9%83%D9%8A%D8%A9-%D9%85%D9%86-%D8%A7%D9%84%D8%A5%D9%83%D9%85%D8%A7%D9%84-%D8%A7%D9%84%D8%AA%D9%84%D9%82%D8%A7/?srsltid=AfmBOoqRMqkvZU7PnCvqtP-

AXMQzjvkz2IcMRFBs4mUnaTHsUfW7ohVP. Published on February 13, 2026 by Mohammed Al-Matari. [Accessed: 03-04-2026, Time: 09:00 AM].
[6] [Online] Available: https://www.21kschool.com/in/blog/digital-education-vs-traditional-education/.[Accessed: 10-04-2026, Time: 11:00 AM].
[7] Rina Zviel-Girshin,Research," The Good and Bad of AI Tools in Novice Programming Education ", Center in Technological and Engineering Education, Faculty of Engineering, Rubin Academic Center, Kfar Monash 4025000, Educ. Sci. 2024, 14, 1089. https://doi.org/10.3390/educsci14101089.
[8] M. Messer, N. Brown, and M. Kölling, "Automated Grading and Feedback Tools for Programming Education: A Systematic Review," ACM Transactions on Computing Education, vol. 24, no. 1, 2023. doi: 10.1145/3636515.
[9] J. Katona and K. I. K. Gyonyoru, "AI-based Adaptive Programming Education for Socially Disadvantaged Students: Bridging the Digital Divide," TechTrends, vol. 69, pp. 925–942, 2025. doi: 10.1007/s11528-025-01088-8.
[10] F. Ali, A. Ahmed, M. A. Alipour, and H. Terashima-Marin, "Adoption of AI-coding assistants in programming education: exploring trust and learning motivation through an extended technology acceptance model," Journal of Computers in Education, 2025. doi: 10.1007/s40692-025-00375-w.
[11] P. Manorat, S. Tuarob, and S. Pongpaichet, "Artificial intelligence in computer programming education: A systematic literature review," Computers and Education: Artificial Intelligence, vol. 8, art. 100403, 2025, doi: 10.1016/j.caeai.2025.100403.
[12] J. Nathaniel, S. S. Oyelere, J. Suhonen, and M. Tedre, "Literature Review on the Integration of Generative AI in Programming Education," International Journal of Artificial Intelligence in Education, vol. 35, pp. 2724–2755, 2025, doi: 10.1007/s40593-025-00524-3.
[13] Y. Xue, H. Chen, G. R. Bai, R. Tairas, and Y. Huang, "Does ChatGPT Help With Introductory Programming? An Experiment of Students Using ChatGPT in CS1," in Proceedings of the 46th International Conference on Software Engineering: Software Engineering Education and Training, 2024, doi: 10.1145/3639474.3640076.
[14] Y. Jing, H. Wang, X. Chen, and C. Wang, "What factors will affect the effectiveness of using ChatGPT to solve programming problems? A quasi-experimental study," Humanities and Social Sciences Communications, vol. 11, 2024, doi: 10.1057/s41599-024-02751-w.

