Journal of Electronic Research and Application, 2025, Volume 9, Issue 3
http://ojs.bbwpublisher.com/index.php/JERA
ISSN Online: 2208-3510
ISSN Print: 2208-3502

AI-Enabled Computer Programming Course Teaching Reform

Yao Dai*

Harbin Institute of Information Technology, Harbin 150431, Heilongjiang, China

**Author to whom correspondence should be addressed.*

Abstract: With the rapid development of modern science and technology, the era of artificial intelligence has quietly come. Against the background of the new era, students' learning needs, learning resource acquisition methods, teachers' teaching concepts, teaching tools, and so on have changed significantly. How to carry out teaching reform based on this change has become one of the important issues facing educators, and the same is true for the teaching of computer programming courses. This paper focuses on the teaching reform of AI-enabled computer programming courses, analyzes its basic problems, and puts forward corresponding reform countermeasures to provide a useful reference for front-line teachers.

Keywords: AI; Computer programming

Online publication: 6 June, 2025

1. Introduction

With the progress of modern science and technology and the rapid development of information technology, programming has gradually become a major cultural accomplishment pursued by people. And with the exponential development of digital technology, programming ability has now become the core component of digital literacy. According to the "2023 Global Programming Education White Paper," most enterprises regard programming ability as the core requirement of technical positions. At present, the development of the current programming course is not satisfactory, some problems need to be solved urgently, such as lagging knowledge update, low value of practice transformation, and insufficient attention to the cultivation of students' thinking. These contradictions are not conducive to the smooth development of the course [1]. The development of AI is conducive to solving the above problems, and provides a new paradigm for it, which has the characteristics of content generation, intelligent diagnosis, personalized guidance, and other aspects, and can provide new ideas for the teaching reform of computer programming courses.

2. Necessity of AI empowering the teaching reform of computer programming courses

The seamless fusion of artificial intelligence technology with higher vocational education has emerged as the primary pathway for educational reform. In this context, the demand for digitally skilled talents across industries continues to rise, with an annual gap reaching up to 2 million individuals, as reported by the Ministry of Human Resources and Social Security. Educators are encouraged to leverage AI technologies to reconstruct the educational framework. Given its robust capabilities in data processing and analysis, AI has become a pivotal instrument for enhancing the optimization and allocation of teaching resources [2]. For instance, one higher vocational institution, by integrating AI technologies, not only expanded its repository of educational materials but also better aligned its offerings with industry requirements. Additionally, students' ability to apply professional knowledge in practical scenarios has markedly improved.

The interactive features of AI technology can facilitate the reform of computer programming courses. For instance, an intelligent system leveraging learning analytics can gather 18 types of learning behavior data in real time, including code debugging frequency and error categories. To illustrate, consider an intelligent platform where data indicates that after teachers adjust their teaching methods, student participation in class has significantly increased by 35%. Additionally, the personalized resource recommendation system markedly enhances students' learning efficiency. An experiment conducted in a classroom demonstrated that the majority of students were able to achieve their course objectives ahead of schedule [3].

The effective integration of intelligent technology can create a positive feedback loop by combining resource efficiency, tailored learning experiences, and competency development. Implementing AI within computer programming curricula supports the enhancement of students' practical capabilities and project-oriented skills while significantly boosting instructors' teaching effectiveness. As technological advancements progress, an emerging educational paradigm is anticipated: "Smart diagnosis-targeted instruction-creative application." This model aims to improve educational standards and contribute to cultivating individuals who are more attuned to modern requirements [4].

3. Teaching status of computer programming courses in higher vocational colleges

The program design course belongs to the core course of the computer major (accounting for 25–30% of the class hours), which shoulders the important mission of cultivating computational thinking (CT), programming ability (PC), and problem-solving ability (PS). The implementation of the course consists of 5 stages, each of which takes up a different proportion, such as analysis (20%), design (30%), coding (25%), testing (15%), and optimization (10%). The teaching of mainstream programming languages (Python, Java, etc.) faces some shortcomings.

3.1. Rapid knowledge update

In the age of artificial intelligence, there has been significant advancement in both computer and information technologies. Alongside the introduction of new programming languages, innovative concepts, tools, and architectures have also emerged. However, influenced by conventional educational approaches, the teaching materials for computer programming courses in higher vocational colleges fail to keep pace with technological advancements. This results in students acquiring knowledge and skills that are somewhat outdated, with insufficient exposure to the latest industry developments. Such a gap is detrimental to their future career prospects [5].

3.2. Students' lack of practical ability

The experimental data show a significant gap between the passing rate of the theoretical examination (79%) and the excellent rate of project practice (21%). An enterprise's pre-job training found that it takes an average of 45 days for fresh graduates to complete primary development tasks independently. The main weaknesses are: (1) Low efficiency of code debugging (average 3 hours/error), (2) weak algorithm design ability (success rate of complex problem modeling < 35%).

3.3. Lack of attention to the cultivation of students' thinking

One of the educational goals in computer programming courses is to develop students' programming thinking. As a unique form of thinking, it focuses on enhancing students' logical reasoning and creative awareness. However, in the teaching process, influenced by traditional educational concepts, instructors might prioritize theoretical knowledge, overlook the development of students' thinking skills, employ outdated teaching methods, and fail to provide a practical platform for students. Consequently, students' programming thinking may not be fully cultivated [6].

4. Specific application of AI-enabled computer programming course teaching reform

Artificial intelligence technology has formed three core application dimensions in the course reform: Intelligent teaching resource construction, personalized learning support system, and intelligent evaluation feedback mechanism.

4.1. Development and integration of intelligent teaching resources

AI technology creates a dynamic resource generation system, enabling real-time updates and smart filtering of educational content via natural language processing. This system can automatically collect cutting-edge online resources and produce digital materials tailored to various learning situations. Additionally, an interactive programming platform powered by AI offers instant support, including code analysis and error detection, while enhancing the engagement of the learning process through visual responses [7].

4.2. Construction of personalized learning support system

By leveraging learning behavior analysis technology, AI systems are capable of recognizing individual learning traits and creating personalized learning trajectories. For learners with a weaker foundation, it offers graduated practice exercises, while for those with exceptional skills, it formulates advanced challenges. Through intelligent resource recommendations, it achieves dynamic adjustment of the learning process, significantly enhancing focus and the efficiency of knowledge acquisition [8].

4.3. Design of intelligent evaluation and feedback mechanism

The limitations of the traditional homework correction mode have been broken through by AI technology, and the intelligent evaluation system realizes multi-dimensional analysis of the code: Not only detecting the normative grammar, but also evaluating the efficiency of the algorithm, code style, and other higher-order indicators. The system provides personalized learning suggestions based on error pattern analysis, forming a closed-loop improvement mechanism of "practice-diagnosis-reinforcement."

5. Teaching innovation strategies of AI-enabled programming courses

The intelligent education environment forms a ternary interactive structure of "teacher-student-agent," and the man-machine synergistic effect is significantly better than that of a single subject. According to the challenges of programming teaching and the educational characteristics of AI, four innovative strategies are constructed. The model presents a bi-directional driving relationship: Teaching challenge stimulates technical potential, and AI characteristics feed teaching innovation, forming a virtuous cycle of "challenge-potential-strategy."

The AI programming function offers technical assistance for innovative teaching methods, fostering programming interest and encouraging innovative thinking by creating personalized learning environments and implementing intelligent evaluation systems [9].

The specific implementation path includes intelligent assisted teaching, automated code generation, interactive programming environment, project-based learning, and other strategies, forming a complete chain of "technology enabling-ability cultivation-value creation," effectively improving teaching quality and students' career competitiveness.

5.1. Promoting curriculum innovation based on intelligent teaching platform

Each student is a unique individual, exhibiting significant variations in learning foundation, habits, and abilities. To effectively address these differences, AI technology can be employed to develop an intelligent teaching platform. Supported by this technology, the platform demonstrates advanced capabilities in personalization and adaptability. By optimizing teaching resources and implementing tailored, tiered instruction, it expands students' learning approaches, encouraging them to move beyond teacher dependency and embrace independent, collaborative, and inquiry-based learning as primary methods. Additionally, the platform fully incorporates emerging technologies, such as virtual reality and artificial intelligence, to diversify instructional strategies, offer innovative learning experiences, and enhance overall course quality [10].

5.1.1. Providing personalized learning experience for students

AI can create tailored and individualized learning paths along with teaching materials for students by analyzing their learning data and behaviors, including learning styles, interests, and abilities, among others. This approach caters to the diverse learning requirements of students, facilitating flexible, adaptive, and personalized learning experiences. Such methods are advantageous in fostering learning interest and boosting motivation, enabling students to engage in learning more enjoyably. Additionally, AI can smartly suggest programming exercises that match the student's progress and skill level, offer customized guidance, and deliver instant feedback, thereby assisting students in acquiring programming knowledge more effectively [11].

5.1.2. Carrying out intelligent tutoring and actively making use of its feedback function

AI can analyze students' code by itself and give a real-time grammar check, error prompt, and optimization proposal. This is helpful for students to detect and correct problems in time when writing code, to improve the efficiency and quality of programming.

5.1.3. Guiding students to self-study and actively explore

AI can provide students with a wealth of programming examples, exercises, and cases to help them consolidate and extend knowledge through practice. Under the guidance of AI, students can complete programming tasks independently, explore different solutions, and share and exchange experiences with other students.

5.1.4. Carrying out intelligent assessment and making grade prediction

AI can automatically evaluate students' programming work and performance, provide objective and accurate grade feedback, automatically analyze students' programming ability and potential, and give personalized learning suggestions and development paths.

5.2. Breaking through grammar troubles and training students' programming thinking

In the process of programming learning, grammar learning is very important, and it is an important foundation. However, excessive focus on grammar may hinder the cultivation of students' programming thinking.

5.2.1. Code generation and proposal

Using an AI-powered code completion tool, students can receive grammar suggestions and automatic completions while writing code. This helps minimize grammar errors and allows students to concentrate more on problem-solving logic. AI is capable of generating relevant code snippets based on students' programming requirements and standards. Students can then refine these snippets, enhancing both their programming efficiency and skills. Additionally, by comparing the AI-generated code with their own, students can improve their programming techniques and styles. With the assistance of generative artificial intelligence, complex programming syntax can be translated into more comprehensible abstract concepts, enabling students to grasp programming logic without directly engaging with specific grammatical rules. By offering advanced programming environments or tools, students can work at higher levels of abstraction, thereby reducing the need to focus on low-level syntactical details [12].

5.2.2. Paying attention to programming teaching and injecting interactive genes

Create interactive programming instruction scenarios that enable students to perform hands-on operations within an accessible programming environment. Through practical engagement, students can gain a deeper comprehension of programming concepts and logical reasoning, ultimately enhancing their programming proficiency and expertise. Facilitate real-time collaboration and interaction among students, as jointly writing code, exchanging insights, and addressing challenges can foster the development of teamwork and communication abilities. AI systems can analyze students' code to deliver evaluations and recommendations. Furthermore, these systems can tailor the instructional material and complexity based on students' learning progression and outcomes, thereby increasing the efficiency and specificity of the teaching process [13].

5.2.3. Flexibly adopt a variety of teaching methods

Employ teaching approaches such as heuristic learning, group discussions, and inquiries to enhance students' interest and engagement in learning, while assisting them in comprehending and acquiring programming concepts. In the event of challenges during experiments, generative AI can offer specific error feedback and direct students toward solutions. Encourage learners to create their algorithms and utilize the analytical resources from generative AI to assess and refine these algorithms effectively.

5.2.4. Scientific programming task design

Design learning tasks that are problem-solving focused and challenging, involving the resolution of real or simulated problems. This allows students to practice programming thinking and enhance their problem-solving skills as they work through these tasks. Engage students by sparking their interest and motivation, encouraging

deeper thought and exploration. Providing a series of programming cases for students to analyze and understand the code logic in the cases is beneficial for building programming thinking [14].

6. Conclusion

In short, the promotion of AI-enabled computer programming course teaching reform is not achieved overnight, nor will it be smooth sailing, and relevant workers need to work for a long time and make continuous efforts [15]. This paper takes AI as the starting point to explore the way of teaching reform of computer programming courses, which is conducive to improving the efficiency and effectiveness of talent training while improving the teaching quality of the course.

Disclosure statement

The author declares no conflict of interest.

References

[1] Zhang Y, Tang L, Ma C, 2023, Research on Paths and Countermeasures of Artificial Intelligence Boosting Teacher Development. Audio-visual Education Research, (10): 104–111.

[2] Jiang L, Wang Z, Ge S, 2023, Discussion on Integrating Artificial Intelligence Technology into Embedded System Curriculum Teaching. China Modern Educational Equipment, (17): 10–12.

[3] Lu J, Ren H, Gu D, et al., 2020, Student-centered Hierarchical Experimental Teaching Reform of Programming. Computer Education, (11): 174–178.

[4] Cai T, 2023, Research on Problem-solving Teaching Mode based on Computational Thinking, dissertation, Northwest Normal University.

[5] Li J, Tang N, Yu J, 2022, Exploration on Multi-level Backward Inquiry Teaching of Artificial Intelligence Programming Course Based on Practice. Computer Education, (10): 145–149.

[6] Deng W, Yang X, Gao Q, et al., 2023, Research on Classroom Teaching Evaluation Model Supported by Artificial Intelligence. China Educational Informatization, 29(8): 3–14.

[7] Bian J, Cao H, Liu L, 2023, Analysis of Personalized Teaching Model based on Big Data Analysis. Integrated Circuit Applications, 40(8): 362–363.

[8] He Y, Wang C, 2020, Flipped Classroom Teaching Practice of C Language Programming based on MOOCs. Computer Education, (1): 160–163.

[9] Deng Z, Li F, Chen X, et al., 2022, Reform of the Teaching Mode of Programming Courses by Strengthening the "High-levelness, Innovativeness and Challengingness." Computer Education, (11): 82–86.

[10] Li Q, Kang J, Feng J, 2023, Analysis of Hierarchical Teaching Strategies in Programming Curriculum. Applications of Integrated Circuits, 40(7): 202–203.

[11] Yu Y, 2024, Research on Teaching Reform and Practice of Artificial Intelligence Empowerment C++ Programming Course. Computer Knowledge and Technology, 20(35): 178–180.

[12] Qin T, 2021, Characteristics of Vocational Students' Learning Ability and Ways of Improvement in the Context of Artificial Intelligence Education. China Vocational and Technical Education, (20): 88–92.

[13] Yang X, 2024, Exploration on Teaching Reform of New Engineering Information Majors based on Artificial

Intelligence Technology. Wireless Internet Technology, 21(2): 107–110.

[14] Zhang Y, 2024, Research on Teaching Reform based on Python and Artificial Intelligence for Electrical Majors. China Educational Technology Equipment, (4): 62–64.

[15] Mai L, Fan M, 2022, Analysis on Professional Construction of Higher Vocational Colleges in the Era of Artificial Intelligence. Vocational and Technical Education, 43(4): 46–52.