

Article

Research on Cultivating Practical Innovation Capability in the Java Web Application Development Course under the "AI Co-creation + Contest-Education Integration" Model

Xiaoyang Sun¹ and Yuxiang Hou^{1,*}

¹ School of Computer Science and Artificial Intelligence (Industrial Software), Guangdong University of Science and Technology, Dongguan, China

* Correspondence: Yuxiang Hou, School of Computer Science and Artificial Intelligence (Industrial Software), Guangdong University of Science and Technology, Dongguan, China

Abstract: Artificial intelligence technologies represented by large language models are profoundly changing the software development paradigm and posing new challenges to the cultivation of software talent. As a core course for computer majors in application-oriented universities, Java Web Application Development generally suffers from outdated content, inauthentic practical projects, insufficient innovation training and a single assessment method. This paper couples the concept of AI co-creation with the mechanism of contest-education integration and constructs a dual-wheel-driven model for cultivating practical innovation capability. It presents the model framework at the levels of goal system, collaborative subjects and implementation path, and elaborates the implementation route from four dimensions: curriculum reconstruction, teaching method reform, a three-tier project gradient and a multi-dimensional evaluation system. A one-academic-year controlled teaching experiment shows that the experimental class significantly outperforms the control class in course grades, practical innovation capability scores and contest awards, with particularly marked gains in innovative thinking and problem solving. The model effectively improves teaching quality and students' practical innovation capability and can serve as a reference for similar course reforms.

Keywords: Artificial Intelligence; AI Co-creation; Contest-Education Integration; Java Web Application Development; Practical Innovation Capability

1. Introduction

Intelligent programming tools such as GitHub Copilot, Tongyilingcode and CodeGeeX have been widely used in the software development industry, which brings double challenges to software courses in colleges and universities. The teaching content with the core of grammar training and handwritten code is difficult to match the real industry scene, resulting in a gap between what is learned in class and what is required in the job. Inappropriate use of AI tools can also foster dependency and weaken independent thinking and problem-solving skills. At the same time, subject competition should be a bridge connecting classroom and industry practice, but most colleges and universities still have the "two skins" phenomenon of competition and teaching, and the channel of transforming competition resources into courses is not smooth. Therefore, this study combines AI collaborative learning with competition education function, integrates AI tool application ability and competition actual combat training into the whole process of course teaching, and focuses on answering three questions: what practical ability and innovation literacy should be cultivated in the AI era curriculum; How to realize the coupling of AI co-creation and competition teaching rather than simple superposition at the mechanism level; How the implementation effect of this model can stand the empirical test. Theoretically, this study integrates the AI collaborative learning theory and the

Received: 21 July 2026

Revised: 11 September 2026

Accepted: 22 September 2026

Published: 25 September 2026



Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

concept of competition and teaching integration, and provides an analytical framework for the innovation of software courses teaching mode in the intelligent age. In practice, operable curriculum reconstruction scheme, project-based teaching path and multiple evaluation system were formed. The research adopts literature research, questionnaire survey, controlled experiment, action research and case analysis, and follows the idea of "raising questions, analyzing problems, constructing patterns, practical verification and summarizing and improving". The overall framework is shown in Figure 1.

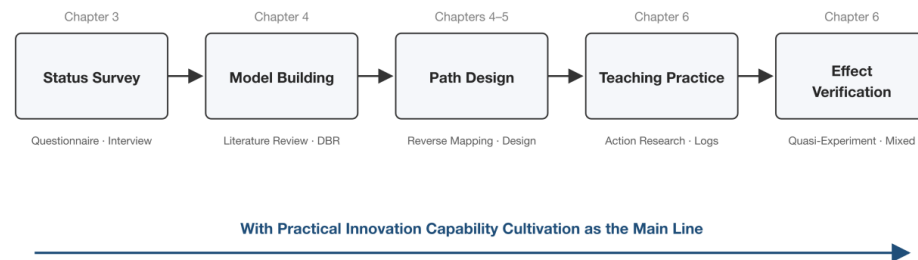


Figure 1. Overall Research Framework.

2. Definition of Core Concepts and Theoretical Basis

2.1. Definition of Core Concepts

AI co-creation is a new teaching form in which teachers, students and artificial intelligence tools collaborate to participate in the whole process of software development: As a collaborative developer, AI is deeply involved in requirements analysis, system design, coding and testing, and students are transformed from code writers to AI commanders and reviewers of achievements. Their core capabilities are to accurately describe requirements, reasonably decompose tasks, critically review generated content, and innovate and iterate. This shifts the focus of learning from repetitive coding to higher-order activities such as architecture design, requirements understanding, and creative thinking. The integration of competition and teaching systematically integrates the concepts, topics, standards and training methods of discipline competitions into curriculum teaching, forming a closed loop of "promoting teaching, promoting learning, promoting innovation and promoting reform by competition". Teaching provides talents for competition, and competition in turn provides teaching with real project carrier and advanced evaluation standards. Both of them jointly serve the cultivation of practical innovation ability -- this study defines it as the ability of students to comprehensively use professional knowledge in real or simulation development situations to complete problem discovery, problem analysis, scheme design, hands-on implementation and iterative optimization, which specifically includes four dimensions of engineering practice, problem solving, innovative thinking and team collaboration. And it is used as the basis of teaching goal setting and effect evaluation.

2.2. Theoretical Basis

Constructivism holds that knowledge is constructed by learners in context with the help of others; project-based teaching and contest practice are effective ways to create authentic contexts for such construction. The conceive-design-implement-operate process advocated by CDIO provides methodological support for the three-tier project gradient. Outcome-based education requires courses to be designed backwards from learning outcomes and is the direct basis for a multi-dimensional evaluation system. Human-AI collaboration theory clarifies the complementary capabilities of people and AI: AI amplifies capability rather than replacing it, and human subjectivity and creativity remain central.

3. Analysis of the Current Teaching Situation and Problems

3.1. Main Existing Problems

At the beginning of the curriculum reform, the research group conducted a survey on the students and teachers of this course from the dimensions of teaching content, practical links and assessment methods. The results showed that there were many shortcomings in the course teaching: theoretical teaching and confirmatory experiments were dominant, and comprehensive project training was obviously insufficient. It is common for students to use AI tools to complete homework, but there is a lack of necessary guidance and norms. Most students believed that there was a big gap between what they learned in the classroom and the future job demand, and they were not confident in their own innovation ability. According to the comprehensive investigation, the problems of course teaching are reflected in four aspects. First, teaching content lags behind technological development. The course is still mainly based on traditional Servlet, JSP basic syntax and simple framework application, and there is little involved in the reasonable application of industry mainstream technologies such as front-end separation, microservice architecture and AI programming tools, and the course content is disconnected from the real development environment of the industry. Second, practice projects lack authenticity and hierarchy. Most of the experimental projects are small verification experiments supporting textbooks, which are isolated from each other and lack complete business logic, so it is difficult for students to experience the complete process of real software development. Third, students' innovation ability training is insufficient. The classroom is dominated by teacher demonstration and student imitation. Students are in a passive state of acceptance, lack of opportunities to face open problems independently, and it is difficult to stimulate critical thinking and innovation consciousness. Fourth, the course assessment method is single. The final written examination or single project acceptance can not truly reflect students' engineering practice level and innovation potential, and it is difficult to guide students' learning behavior to change to ability-based.

3.2. Problem Attribution Analysis

There are deep reasons for the above problems. From the perspective of course orientation, the long-term positioning of the course is a knowledge improvement course rather than a ability training course, which leads to the teaching organization around knowledge points rather than ability goals. From the perspective of resource conditions, teachers lack the accumulation of enterprise front-line development experience and competition guidance experience, and it is difficult to design project tasks close to the real, and the collaborative education mechanism between schools and enterprises is not perfect. From the perspective of technology impact, the rise of AI programming tools has questioned the value of the course. Some teachers have a negative avoidance attitude toward students using AI to write code, and fail to actively transform AI application ability into a new teaching goal, but make the classroom far away from students' real learning behavior. These problems are intertwined, and it is difficult to solve them by only local repair. It needs systematic reconstruction from the level of training mode.

4. Connotation and Construction of the Model

4.1. Analysis of Model Connotation

The model can be summarised as follows: taking the cultivation of practical innovation capability as the goal, AI co-creation as the technical means and productivity support, and contest-education integration as the organisational carrier and driving mechanism, so as to achieve a two-way coupling in which AI empowers contest training and contests steer AI application. AI co-creation addresses how to develop efficiently and learn individually: students use AI to build prototypes quickly and obtain instant feedback, devoting more energy to higher-order creation. Contest-education integration addresses what to learn and to what level: contest problems and standards bring frontier industrial needs into the classroom and provide authentic, challenging tasks for AI collaboration. The two are mutually conditional: without AI co-creation, the efficiency and

coverage of contest training cannot improve; without contest-education integration, AI application training degenerates into tool demonstration for lack of real tasks.

4.2. Model Construction Framework

The pattern framework includes three levels: goal, subject and implementation path: The systematic differences between the old and new teaching models are shown in Figure 2, and the overall framework of the "AI co-creation + competition education integration" dual-wheel-driven mode is shown in Figure 3. At the goal level, a three-layer progressive goal system of "knowledge-ability-innovation literacy" was established. The knowledge layer mastered the core technologies and principles of Java Web development, the ability layer had the ability to use AI tools to efficiently complete the development of complete projects, and the innovation literacy layer formed problem awareness, critical thinking and innovative spirit. At the subject level, a four-party collaborative education community of "teacher-students-AI-competition platform and enterprise" was formed, in which teachers were task designers and learning guides, students were project masters and AI commanders, AI was collaborative developers and instant mentors, and competition platforms and enterprises provided real needs and evaluation standards. At the implementation path level, a five-dimensional linkage implementation system of "curriculum reconstruction, method reform, project gradient, evaluation reform, guarantee construction" was formed, which jointly served the core goal of practical innovation ability training. Compared with the traditional teaching model, this model has achieved four fundamental changes: The teaching goal changes from mastering knowledge to cultivating ability and innovative quality, the relationship between teachers and students changes from teaching and imitation to AI co-creation between teachers and students, the teaching content changes from textbook knowledge points to competition questions and real projects, and the evaluation method changes from the final examination to the whole process, diversified and value-added evaluation. The systematic differences between the new and old teaching modes are shown in Figure 2. Figure 3 shows the overall framework of the two-wheel drive mode of "AI co-creation + competition and teaching integration".

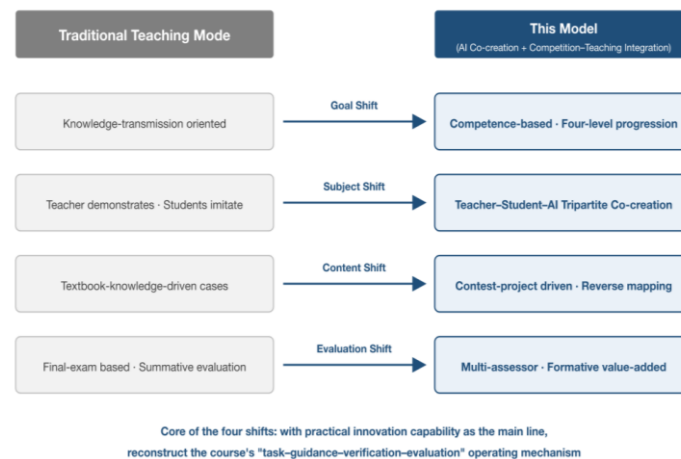


Figure 2. Comparison between the Traditional and the New Teaching Mode.

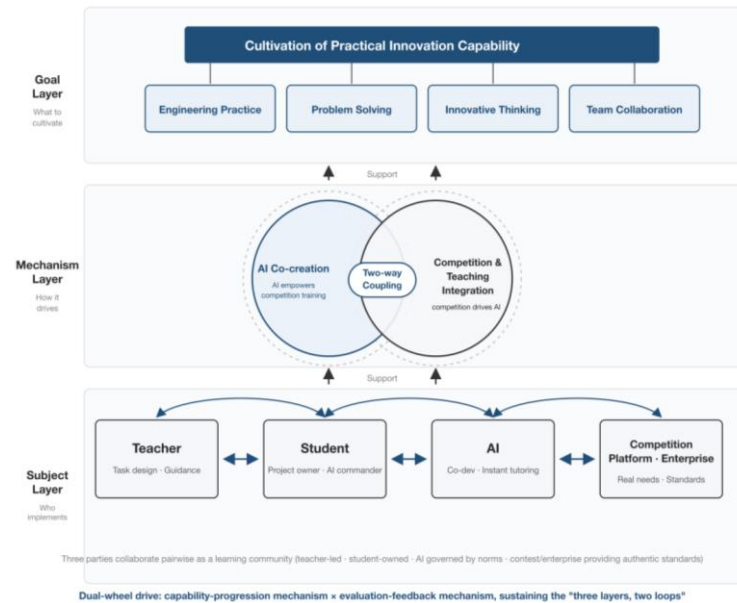


Figure 3. Overall Framework of the "AI Co-creation + Contest-Education Integration" Model.

4.3. Implementation Path of the Model

4.3.1. Reconstruction of the Curriculum System and Teaching Content

The reconstruction follows the principles of backward design and project leadership. Benchmarking the Java tracks of the National College Computer Design Competition, the Internet Plus Innovation and Entrepreneurship Competition and the Blue Bridge Cup, together with real enterprise job requirements, the study derives a core capability map and reorganises the content accordingly. The content is divided into three modules: a basic technology module retaining essential principles; a project practice module using adapted contest problems and real enterprise cases to carry the main capability training; and an AI collaboration module teaching the use of AI programming tools, prompt engineering, review and debugging of AI-generated code, and academic integrity norms, so that students form a responsible attitude towards technology use.

4.3.2. Reform of Teaching Methods Driven by AI Co-Creation

Human-machine pair programming is promoted throughout teaching: one student decomposes requirements and designs prompts, while the other reviews, tests and optimises the AI-generated code, with roles rotated regularly so that every student experiences both commanding AI and mastering code. The teacher's role shifts from code demonstrator to task designer, process observer and trouble shooter, guiding students to understand the principles behind AI-generated code and to identify its defects and risks, so that they can use, discern, correct and innovate. The instant feedback of AI tools also supports an individualised rhythm of learning by doing and learning from mistakes, giving students of different levels appropriate challenges and easing the dilemma that strong students are under-challenged while weak students cannot keep up. The implementation process of this pair programming teaching is shown in Figure 4.

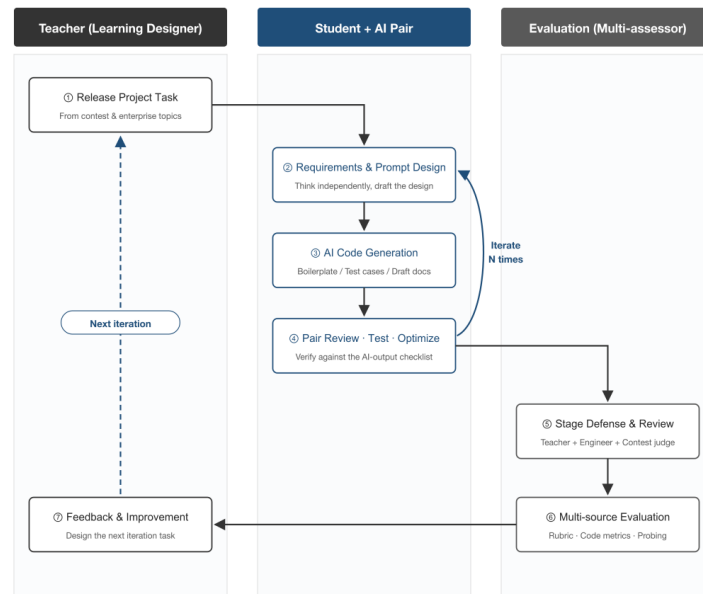


Figure 4. Teaching Implementation Flow of AI Co-Creation.

4.3.3. Project-Based Practice Design of Contest-Education Integration

A three-tier project gradient of classroom projects, contest projects and enterprise projects is constructed. Classroom projects face all students and are simplified adaptations of contest problems running through the whole semester, so that students build contest knowledge while completing them. Contest projects face students with spare capacity and connect directly to official contests, forming a mechanism of training in class and competing outside it. Enterprise projects introduce real business requirements through school-enterprise cooperation as advanced extension, giving outstanding students access to real production environments. After each contest, new problems, technologies and excellent solutions are fed back into the classroom project library, forming a dynamically updated closed loop of teaching resources. The progressive relationship of the three-tier project gradient is shown in Figure 5.

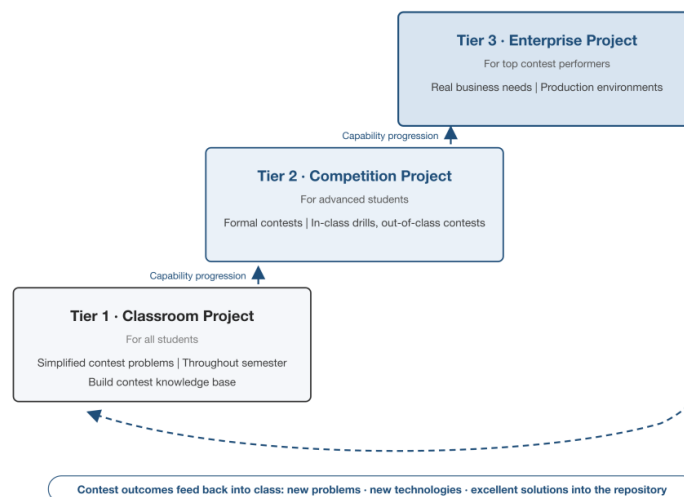


Figure 5. Three-tier Project Gradient.

4.3.4. Construction of a Diversified Assessment and Evaluation System

Following outcome-based education, a three-dimensional system of process, outcome and value-added assessment is established. Process assessment accounts for 40 per cent and examines project iteration, records of AI use (prompt quality and code review

ability), teamwork and stage defences. Outcome assessment accounts for 40 per cent and centres on the final comprehensive project, judged by functional completeness, technical innovation, code quality and user experience, with enterprise engineers and contest judges invited to score from an industrial perspective. Value-added assessment accounts for 20 per cent and is based on the improvement in each student's measured practical innovation capability, focusing on real individual growth. The system acknowledges that students develop with AI while preventing capability dilution caused by AI doing the work for them, balancing encouragement of technology use with adherence to capability-based standards.

5. Teaching Practice and Effect Analysis

5.1. Teaching Practice Design

Two parallel classes in the authors' university were selected for a controlled teaching experiment: the experimental class was taught with the new model and the control class with the traditional mode. The two classes showed no significant difference in entry scores or professional background, and the teachers and teaching hours were kept identical. The practice lasted one academic year, covering two rounds of teaching. During the process, classroom observation logs, learning-platform data and structured interviews were used to collect process data continuously, ensuring completeness and traceability.

5.2. Evaluation Indicators and Data Collection

The study builds an evaluation scheme in which multiple data sources corroborate each other. Quantitative data include course examination results, pre- and post-test scores on the practical innovation capability scale, and contest participation and awards. Qualitative data include student interviews, analysis of project work and teachers' reflective logs. The scale was developed around the four dimensions of engineering practice, problem solving, innovative thinking and teamwork, and expert review and a small-sample pilot confirmed good reliability and validity.

5.3. Data Analysis and Results

Statistical analysis shows that the experimental class scored significantly higher in overall course grades and that the proportion of high-scoring students rose markedly. Post-test scores on the practical innovation capability scale show that the experimental class improved significantly more than the control class in innovative thinking and problem solving, and the value-added difference remains significant after controlling for pre-test level, indicating that the improvement does not stem from differences in student intake. Contest participation data show that the experimental class took part more actively, with participation and award rates far above those of the same period in previous years and of the control class. Qualitative analysis further reveals that most students felt AI co-creation enabled them to attempt more complex ideas, that the challenge of contest projects pushed them out of their comfort zone, and that rotating roles in pair programming taught them both to raise requirements and to gate-keep code. Some students admitted a tendency to over-rely on AI, which suggests that guidance and process supervision must not be relaxed. The pre- and post-test changes of the two classes across the four dimensions are shown in Figure 6.

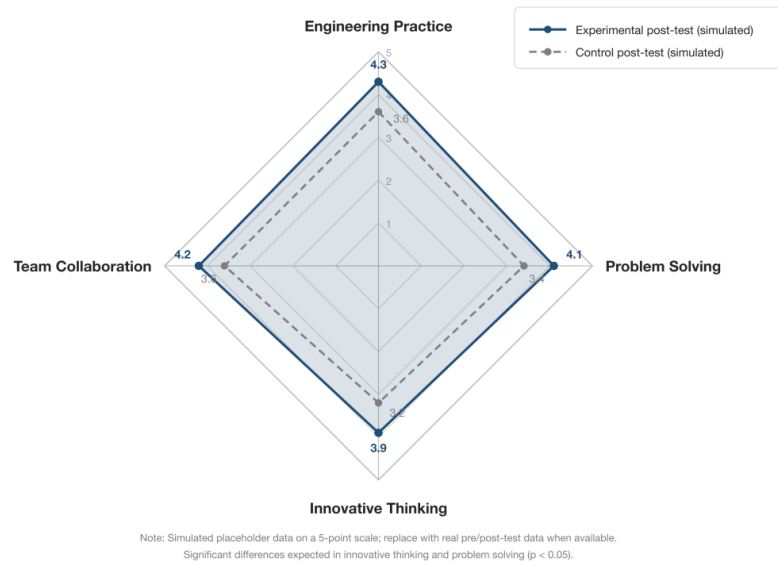


Figure 6. Radar Chart of the Four-Dimension Practical Innovation Capability Assessment.

6. Discussion

Combining the quantitative and qualitative results, the model is clearly effective in stimulating interest, improving practice efficiency and cultivating innovation capability, meeting the expected reform goals. Three shortcomings remain. First, some weaker students tend to become mere porters of AI output and their higher-order thinking is under-trained, so a graded AI-use scaffold is needed. Second, the mechanism for transferring contest resources into the curriculum is incomplete and needs to be more efficient. Third, multi-dimensional assessment increases teachers' workload and requires informatic tools to remain sustainable.

Guarantee Mechanism: Sustainable operation requires support on several fronts. For teaching staff, enterprise practice, contest guidance training and AI teaching workshops are used to improve teachers' dual-professional quality and AI literacy. For platforms and resources, a course platform integrating an AI programming environment, a contest problem bank, a project case library and process evaluation is built. For mechanisms, contest guidance and curriculum reform workload are included in performance appraisal, and academic integrity and AI-use norms are revised to clarify the boundaries and disclosure requirements of AI use. For collaboration, cooperation with software enterprises and contest organisers is deepened to secure stable sources of projects and evaluation resources, moving the educational community from concept to institutional operation.

7. Conclusion

Addressing the new challenges facing the Java Web Application Development course in the AI era, this study constructs a dual-wheel-driven model of AI co-creation and contest-education integration for cultivating practical innovation capability, and provides a complete implementation path across curriculum system, teaching method, project design and evaluation reform. The controlled experiment shows that the model significantly improves learning outcomes and practical innovation capability and strengthens contest participation and innovation confidence, and is therefore worth promoting. Its core contribution is the coupling mechanism between AI co-creation and contest-education integration, which clarifies the value transformation of software courses from writing code to innovating with AI.

The study has limitations: the sample comes from only two classes in one institution, so external validity needs strengthening; the practice lasted only one academic year, so long-term effects are unverified; and innovation capability was measured mainly through

scales and work evaluation. Future research can carry out multi-round, cross-institution verification, apply learning analytics to the AI collaborative development process, and explore transferability to other programming courses.

Funding: This research was supported by the Project of Teaching Quality and Teaching Reform of Undergraduate Universities in Guangdong Province in 2025 (serial number: 338) and the Quality Project of Guangdong University of Science and Technology (project number: GKZLGC2025020).

References

1. J. Chen, K. Li, K. Li, et al., "Dynamic planning of bicycle stations in dockless public bicycle-sharing system using gated graph neural network," *ACM Transactions on Intelligent Systems and Technology*, vol. 12, no. 2, pp. 1–22, 2021.
2. H. Liang, X. Zhan, F. Ren, et al., "Empirical research on the Talent Training Mode of new Engineering Competition and Education Integration," *Journal of Hubei Normal University (Natural Science Edition)*, vol. 46, no. 2, pp. 113–118, 2026.
3. M. Wang and Y. Guo, "Practice and exploration of 'Integration of Competition and teaching' in Higher Vocational Colleges," *Education Theory and Practice*, vol. 45, no. 27, pp. 60–64, 2025.
4. L. Jiang, H. Huang, and J.-j. Liu, "The cultivation of innovation and practice ability of artificial intelligence professionals by 'Tutor traction + Competition drive'," *Computer Education*, no. 4, pp. 220–224, 2023.
5. X. Jia, "Exploration on Teaching Reform of C Language Programming Course in the Era of Artificial Intelligence," *Information and Computer*, vol. 37, no. 24, pp. 229–232, 2020.
6. M. Zhao, Z. Zhang, and X. Wang, "Experimental Teaching Reform and Practice of Programming Courses Driven by Generative AI+ Project," in **Proceedings of the 33rd National Conference on Computer New Technology and Education**, National Association for Computer Education in Higher Education, 2025, pp. 511–516.
7. X. Su and Q. He, "Exploration of the Innovation of Teaching and Learning Scenarios and Modes Empowered by Artificial Intelligence — Taking the Programming Course as an Example," *Chinese University Teaching*, no. 6, pp. 65–72, 2025.
8. Z. Zhu, L. Dai, and J. Hu, "High Awareness Generative Learning: The Innovation of Learning Paradigm Empowered by AIGC Technology," *Audio-visual Education Research*, vol. 44, no. 6, pp. 5–14, 2023.
9. F. Wu, Z. Xia, and S. Gao, "Reshaping Smart Learning Environment with Generative Artificial Intelligence: From Factor Improvement to Ecological Reconstruction," *Research on Audio-visual Education*, vol. 46, no. 1, pp. 54–63, 2025.
10. X.-H. Luo, "Generative AI-Enabled Interactive Teaching: The Construction and Practice of the 'Human-Machine Cooperation-Project-Driven' Model: A Case Study of the Course of Fusion Media Content Production," *University Education*, no. 7, pp. 164–168, 2026.
11. Y.-w. Zhang and S. Tao, "Exploration on Teaching Reform of Web Front-end Design Course Based on Enterprise Project Practice," *Computer Knowledge and Technology*, vol. 18, no. 29, pp. 165–167, 2012.
12. W. Liu, Y. Zhou, X. Ma, et al., "Project-based Teaching Design and Practice of Java Web Comprehensive Practice Course Driven by Virtual Simulation and LLM Collaboration," *Computer Education*, no. 7, pp. 189–196, 2026.
13. Y. Wang, F. Ru, X. Jin, et al., "Teaching Reform and Application Research of Web Front-end Development Course Based on AI-Driven," *Computer Times*, no. 9, pp. 77–83, 2026.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of Publisher and/or the editor(s). Publisher and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.