ISSN 2663-8169 Vol. 8, Issue 10: 64-69, DOI: 10.25236/IJNDE.2026.081008

Exploration and Practice of Competence-Oriented Teaching Reform in the C Language Programming Course

Yan Jiang[1,2,a], Xiufang Li[1,2,b], Wenpeng Lu[2,c,*]

[1]Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan, China
[2]Shandong Provincial Key Laboratory of Computer Networks, Shandong Fundamental Research Center for Computer Science, Jinan, China
[a]jiangy@qlu.edu.cn, [b]lixf@qlu.edu.cn, [c]luwpeng@163.com
[*]Corresponding author

Abstract: *At present, programming language courses face several common problems: theory and practice are disconnected, teaching processes are hard to monitor effectively, students' proficiency levels vary widely, and personalized learning is difficult to achieve. This paper takes the C Language Programming course as an example and proposes a relatively systematic teaching reform scheme to address these problems. The scheme centers on improving students' programming practice competence and overall quality. It adopts a series of measures: optimizing the curriculum, innovating teaching methods, building an online judging system, integrating values-based education into the curriculum, promoting learning through competition, and reforming assessment methods. These measures aim to address the pain points in teaching, stimulate students' interest in learning, and develop their ability to learn independently, their teamwork skills, and their creative thinking. This paper describes the design of the reform scheme, the key problems it aims to solve, its innovations, and its implementation outcomes, with the aim of providing a useful reference for teaching reform in programming courses.*

Keywords: *C language; teaching reform; practical competence; learning through competition; online judging system*

1. Introduction

Programming language courses are professional foundation courses for computer-related majors. They are also important prerequisite courses for core courses such as Data Structures and Operating Systems. Such courses should not only help students understand and master the basics of programming, but also develop their programming ability and good programming practices. They are therefore a key part of achieving the all-round development of knowledge, ability, and quality. Traditional teaching models, however, tend to focus on the transmission of grammatical knowledge. Theory and practice are often disconnected, and students frequently do not know where to start when facing practical problems. In large classes, this problem becomes even more serious: teachers find it hard to monitor students' learning effectively, the code duplication rate of assignments remains high, and the gap between strong and weak students keeps widening. Therefore, programming language courses need in-depth teaching reform. Exploring more effective teaching models has become urgent for improving the quality of computer science talent cultivation.

Many universities in China and abroad have explored this area, and their efforts have been helpful. For example, at Central China Normal University, a case-guided teaching reform based on OBE theory was implemented in the C Language Programming Laboratory course [1], while some foreign universities have adopted the problem-based learning (PBL) approach to encourage students to learn independently and solve problems collaboratively[2]. These practices show that student-centered reform with a strong emphasis on practice has significant advantages. Against this background, the project team selected "C Language Programming", a core foundation course for computer-related majors, as the research object. The team then carried out a round of teaching reform aimed at improving students' programming practice competence and overall quality [3].

ISSN 2663-8169 Vol. 8, Issue 10: 64-69, DOI: 10.25236/IJNDE.2026.081008

This paper first analyzes the current pain points in C language teaching, and then explains the design philosophy and practical pathway of the proposed reform scheme. The scheme unfolds along four dimensions. The first is curriculum content reconstruction. The second is teaching method innovation, including case-based teaching and project-driven teaching. The third is teaching technology empowerment, namely the self-developed online judging system. The fourth is educational mechanism expansion, covering "learning through competition" and the integration of values-based education into the curriculum. By comparing student learning outcomes, practical competence, and competition results before and after the reform, this paper aims to verify the effectiveness of the comprehensive reform scheme and to distill a teaching paradigm for programming language courses that can be replicated and applied elsewhere.

2. Key Problems to Be Solved by the Reform

This teaching reform project aims to address four key problems in current C language teaching:

- First, theory and practice are disconnected, and knowledge becomes fragmented. Traditional lecture-based teaching ensures that knowledge is systematic, but students receive information passively. They do not understand how the grammar they learn fits into a complete program, and their knowledge therefore becomes fragmented. The limited number of experiments and assignments is also not enough to connect theoretical knowledge with practical application.
- Second, teaching processes lack effective monitoring, and assessment methods are limited. Traditional teaching mainly assesses learning outcomes through lab reports and assignments. It cannot monitor how students complete their work, and it is hard to prevent perfunctory work or copying. In large classes of around one hundred students, grading assignments takes a huge amount of time, and teachers have no way to accurately gauge each student' actual level.
- Third, students have uneven proficiency levels, which makes personalized learning difficult to achieve. When classes are combined, the gap in students' programming backgrounds becomes more obvious. A one-size-fits-all approach to content and pace often leaves stronger students under-challenged and weaker ones struggling to keep up, which widens the gap between the two groups.
- Fourth, assessment methods are rigid and cannot fully measure students' abilities. A single final written exam plus regular assignments cannot comprehensively evaluate students' actual programming ability, problem-solving ability, or creative thinking. Nor can it motivate students at different levels to reach their full potential.

3. Design and Practice of the Teaching Reform Scheme

To address the above problems, the project team designed a relatively systematic teaching reform scheme, which was repeatedly adjusted and gradually refined in actual teaching.

3.1 Optimising the Curriculum and Building a Tiered, Progressive Knowledge System

To make the teaching content systematic and complete, and to accommodate students with different starting points, the project team divided the course content into three levels:

- Basic knowledge level: This level targets students with no prior programming experience. It covers data types, operators, expressions, and sequence, selection and loop structures, ensuring that all students can gain a foothold.
- Core knowledge level: This level focuses on core content such as arrays, functions, pointers and structures. Through case-based teaching, students are helped to grasp the principles of modular programming [7].
- Advanced knowledge level: This level is designed for students who have spare capacity. It introduces common algorithms (such as sorting and searching), file operations and basic project development knowledge, guiding them towards deeper exploration.

ISSN 2663-8169 Vol. 8, Issue 10: 64-69, DOI: 10.25236/IJNDE.2026.081008

3.2 Innovating Teaching Methods and Strengthening the Integration of Theory and Practice

- Introducing case-based and project-driven teaching. Classroom instruction abandons tedious grammatical enumeration. Instead, it uses simple, typical teaching cases that capture students' interest. Knowledge is explained under the guidance of needs, so that students understand how what they learn fits into a program. In the course design component, comprehensive projects related to students' real-life contexts (such as student grade management systems and Snake games) are set. Teams of 3–5 students complete these projects collaboratively, cultivating their teamwork spirit and their ability to solve complex problems.
- Integrating values-based education. Values-based content is naturally integrated into teaching. For instance, when explaining algorithms, the rigour and responsibility of programs are emphasised. When discussing programming standards, the spirit of craftsmanship is introduced. In project practice, students are encouraged to use technology to address real social problems. This achieves an organic unity of knowledge transmission and value guidance [6].
- Implementing the flipped classroom with the support of new technologies. Tools such as Rain Classroom are used to push preview materials (such as MOOC videos and study guides) before class. During class, real-time quizzes and bullet comments enhance teacher-student interaction. After class, review questions and extended reading are pushed to students. This breaks the temporal and spatial constraints of the traditional classroom and improves teaching efficiency and effectiveness.

3.3 Independently Developing an Online Judging System to Strengthen Process Monitoring

In response to the difficulties of monitoring in large-class teaching and the heavy burden of grading assignments[4], the project team independently developed a fully functional online judging system. The system offers the following core functions:

- Automatic judging and instant feedback: Students submit code online, and the system immediately compiles, runs, and judges it automatically, providing detailed feedback (such as compilation errors, running result comparisons, and scores), enabling students to intuitively identify problems and make timely corrections.
- Process monitoring and data-driven instruction: This model assigns "slightly advanced" knowledge points through the online OJ platform, combined with high-frequency training of 12–20 problems per week, while incorporating past competition problems and live competition opportunities. The system records behavioral data such as the code submitted by students on each occasion, the number of submissions, and time consumed, providing teachers with detailed learning analytics reports. This enables teachers to accurately grasp each student's learning status and difficulties, thereby delivering targeted personalized guidance.
- Intelligent plagiarism detection to eliminate copying: The system incorporates an efficient code plagiarism detection function that effectively identifies and flags code plagiarism, upholds academic integrity, and encourages students to complete experimental tasks independently.

3.4 Promoting Learning Through Competition and Building a Platform for Students to Demonstrate Their Skills

The project team actively organizes and encourages students to participate in various professional competitions, such as the "Blue Bridge Cup" National Software and Information Technology Professionals Competition and the China Collegiate Computing Contest—Team Programming Ladder Tournament. Through competition-based practice and learning[5], significant outcomes can be achieved in the following four aspects:

- Stimulating learning motivation: The challenge and prestige of competitions greatly stimulate students' competitiveness and intellectual curiosity, shifting them from passive learning to active exploration.
- Sharpening practical skills: Competition problems are often more challenging and comprehensive than regular exercises, effectively exercising students' problem analysis ability, algorithm design ability, and capacity to solve problems within a specified time, while cultivating their psychological resilience under pressure.

ISSN 2663-8169 Vol. 8, Issue 10: 64-69, DOI: 10.25236/IJNDE.2026.081008

- Broadening professional perspectives: Through competitions, students are exposed to more cutting-edge technologies and more outstanding peers, broadening their horizons and accumulating valuable experience and capital for future further study or employment.
- Reforming assessment methods: The reformed assessment system lets students apply to have competition awards count towards their regular grades, or even their lab grades. This gives students with a particular strength a pathway that suits them.

3.5 Strengthening Industry Cooperation and Aligning with Market Demand

The project team has consistently maintained close cooperation with enterprises. Enterprise engineers are invited to participate in the guidance of course design and to deliver technical lectures, and real-world cases from enterprises are also introduced into the classroom. In this way, students are exposed to the latest technological developments in the industry during their studies and gain an understanding of what kind of talents enterprises actually need. As a result, they develop a clearer understanding of their learning objectives, while their professional literacy and employment competitiveness are enhanced.

4. Innovative Points of the Reform Scheme

This reform scheme is not a partial adjustment of a single measure. Instead, it is a systematic design focused on competence development. It covers four dimensions: teaching methods, technical means, assessment system, and educational model. Specifically, the innovation in teaching methods addresses the question of how to teach; the innovation in technological means addresses what tools can support teaching and learning and how the learning process can be monitored; the innovation in the assessment system addresses how learning outcomes can be measured scientifically; and the innovation in the education model answers the fundamental question of what kind of talent is to be cultivated. The four dimensions are distinct in emphasis yet mutually reinforcing and jointly advanced: teaching methods and technological means provide the implementation path and platform support for competence development, the assessment system provides orientation and feedback, and the education model anchors its value orientation. Together, they form an integrated reform framework that extends from the classroom to extracurricular activities, from teaching to assessment, and from knowledge transmission to the cultivation of overall qualities. The main innovations of the scheme are elaborated in the following four aspects:

- Innovation in teaching methods. A diversified teaching model of "case-based classroom teaching + project-driven practice + online system support" has been constructed, organically integrating theory with practice, individual learning with teamwork, and in-class learning with extracurricular competitions. Case-based teaching takes typical problems as its entry point, enabling students to understand the practical use of grammatical knowledge in concrete contexts and thus overcoming the previous disconnection between "grammar first, application later". Project-driven practice requires students to complete comprehensive projects in teams of three to five, experiencing the basic process of software development through the complete cycle of requirement analysis, module design, coding and debugging, and final presentation, thereby strengthening their communication skills and their ability to solve complex problems. The online judging system links pre-class preview, in-class interaction, and after-class training into a closed loop of "teaching, learning, practicing, and assessing", so that students learn by doing, with theoretical study and competence training advancing in step.
- Innovation in technical means. The project team independently developed a fully functional online judging system, which not only enables automated evaluation and instant feedback but also collects and analyzes process data. This system provides strong support for precision teaching and individualized tutoring, and effectively addresses the difficulty of monitoring learning in large classes. It records the compilation results, running status, time consumption, and pass rate of each submission, thereby generating learning behavior data covering the entire class. On this basis, teachers can promptly identify common errors and weak knowledge points, dynamically adjust teaching priorities, and offer targeted individual guidance to struggling students. Functions such as intelligent plagiarism detection, question bank management, and training with past competition problems operate in coordination, integrating previously scattered homework, laboratory tasks, and competition training into a single platform. This substantially reduces teachers' marking workload and provides a reliable vehicle for the routine implementation of learning through competition.

ISSN 2663-8169 Vol. 8, Issue 10: 64-69, DOI: 10.25236/IJNDE.2026.081008

- Innovation in the assessment system. A multiple assessment system of "classroom performance + assignments and experiments + final examination + competition awards" has been established; in particular, the measures of "substituting competitions for routine training and awards for examinations" have fully stimulated students' intrinsic potential and enabled individualized evaluation of students at different levels. This system shifts the focus of evaluation from a single final examination to the whole learning process, with a notably higher proportion of formative assessment, guiding students to make sustained efforts in their daily study. The content of assessment has shifted from knowledge memorization to competence application, with hands-on programming and project defense becoming the main forms of examination, so that students' real programming level can be objectively presented. Meanwhile, stratified and categorized evaluation is implemented: students with a weaker foundation obtain basic marks by completing fundamental tasks, while more able students can earn bonus marks and personalized development channels through competition awards and challenging tasks, ensuring that students of all levels gain a sense of achievement and sustained motivation for learning.
- Innovation in the educational model. Elements such as values-based education, teamwork, and university-industry cooperation have been deeply integrated into the entire teaching process[8], with the aim of cultivating well-rounded talents who possess solid professional skills as well as sound professional ethics and an innovative spirit. While imparting knowledge, value guidance is achieved through the cultivation of programming standards, the rigor of algorithm design, and the fulfillment of project responsibilities, so that values education takes effect imperceptibly. Through team projects and authentic enterprise cases, students gain early exposure to industry trends and professional requirements, thereby narrowing the gap between the classroom and the workplace. Learning through competition also provides a platform for demonstrating competence and fosters students' resilience, teamwork spirit, and innovation awareness. As a result, an educational pattern has taken shape in which the cultivation of professional competence and the development of comprehensive qualities advance in the same direction and reinforce each other, offering a referable paradigm for the teaching reform of similar courses.

5. Implementation Outcomes

This teaching reform scheme has been piloted in several classes, and initial results have been achieved. Figures 1 and 2 show the score distributions of the 2023 control group and the 2025 experimental group, respectively. In practice, the proportion of students with good or excellent grades has risen noticeably. At the same time, students' interest in learning and their practical ability have improved significantly, classroom participation has increased markedly, and assignment copying has dropped sharply. As for academic competitions, first-year software engineering students from the 2024 cohort and earlier generally won only one national-level award. In contrast, our 2025 software engineering students have won more than five national algorithm competition awards across various disciplines. Their teamwork skills and creative thinking have also been effectively developed.

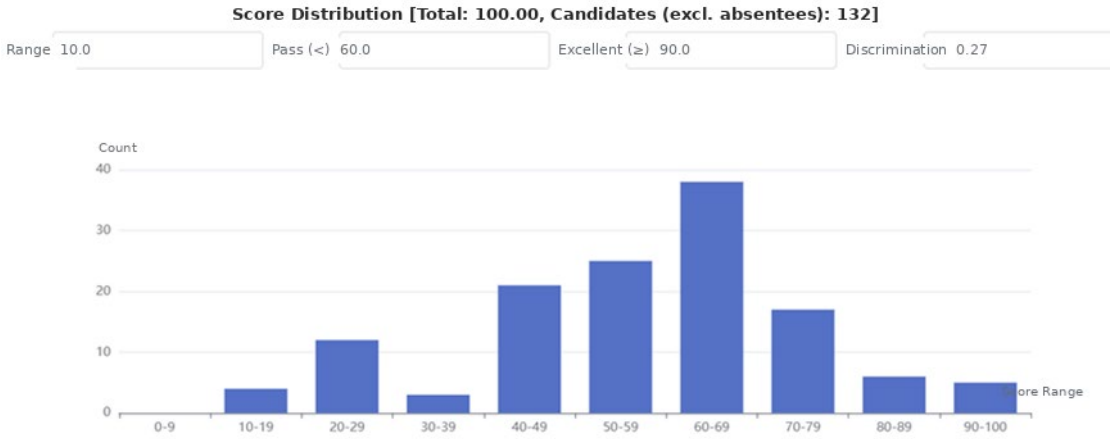


Figure 1: Score distribution of the 2023 cohort (control group)

ISSN 2663-8169 Vol. 8, Issue 10: 64-69, DOI: 10.25236/IJNDE.2026.081008

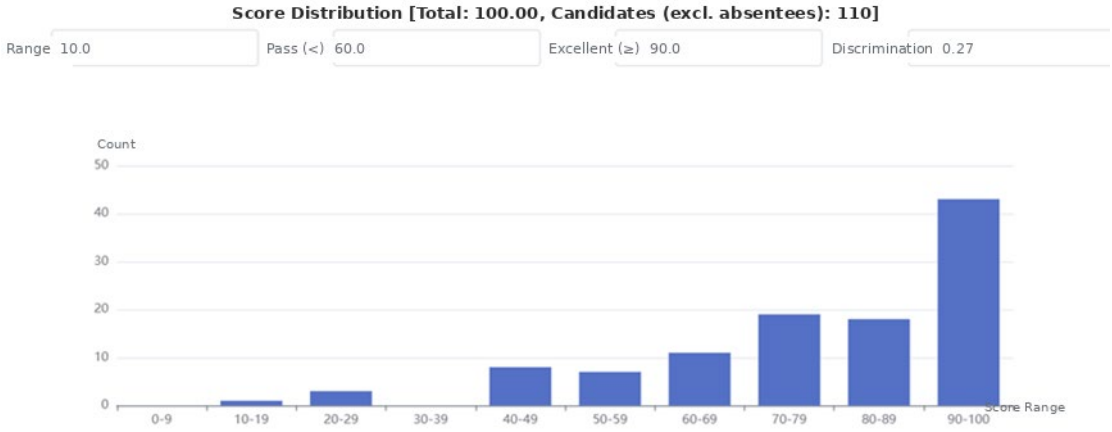


Figure 2: Score distribution of the 2025 cohort (experimental group)

6. Conclusion

This paper takes the C Language Programming course as an example and explores a teaching reform path oriented toward competence development. The reform combines several measures: optimizing the curriculum system, innovating teaching methods, bringing in an online judging system, and promoting learning through competition. Together, these measures effectively address key problems in traditional teaching, such as the disconnect between theory and practice, the lack of process monitoring, and the difficulty of providing personalized learning. Practice shows that the reform can significantly improve students' learning interest, programming practice competence, and overall quality. It therefore carries both theoretical and practical value for advancing teaching reform in programming language courses and improving the quality of talent cultivation. The experience and model can also be applied to similar courses.

References

[1] Liu Yi. Research on teaching reform of the case-guided C Language Programming experimental course based on OBE theory [J]. Experiment Science and Technology, 2024, 22(3): 94–99.
[2] Zhang Wenguang, Zeng Junjie, Ren Jianzhuang, et al. Global status and development trends of teaching methods based on problem-based learning and case-based learning [J]. Journal of Henan Medical College, 2025, 37(3): 346–349. DOI:10.20261/j.cnki.2097-6089.2025.03.088.
[3] Su Xiaohong, Miao Qiguang, Chen Wenyu. Personalized teaching model for enhancing programming competence based on AI empowerment and industry-education integration [J]. China University Teaching, 2023, (6): 4–9.
[4] Zhang Jingjing, Huang Yicheng, Jiang Han, et al. From asynchronous to "quasi-synchronous": Mining and characterizing online behaviors of teachers and students in large-scale blended general education courses [J]. China Educational Technology, 2025, (7): 86–96.
[5] Yan Tao, Gao Chaobang, Gu Musong, et al. Reform of advanced programming courses based on discipline competitions [J]. Computer Education, 2026, (4): 59–64. DOI:10.16512/j.cnki.jsjjy.2026.04.001.
[6] Liu Ying, Fan Qiuhong. Teaching reform of programming courses under "values-based leadership, disciplinary synergy, and AI empowerment"[J]. Computer Education, 2026, (7): 154–160. DOI:10.16512/j.cnki.jsjjy.2026.07.017.
[7] Liu Minghua, Yang Lijuan, Yin Binli, et al. Case-based teaching reform of programming courses oriented toward computational thinking cultivation [J]. Computer Education, 2025, (1): 66–70. DOI:10.16512/j.cnki.jsjjy.2025.01.029.
[8] Xia Huili. Teaching reform and practice of "Programming Fundamentals" oriented toward new engineering education [J]. Journal of Hubei Polytechnic University, 2024, 40(1): 67–71.