World Journal of Educational Studies
ISSN: 2959-9997
DOI: https://doi.org/10.61784/wjes3023

REFORM AND PRACTICE OF TEACHING OBJECT-ORIENTED PROGRAMMING COURSES

YunHui Li
School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, Guangxi, China.
Corresponding Email: 4937717@qq.com

Abstract: Object-Oriented Programming (OOP) is a core course in the computer major. With the rapid development of technology, the teaching content urgently needs to be updated to adapt to new frameworks and tools. This teaching reform systematically innovates the teaching content and methods in terms of the syllabus, teaching design, practical teaching materials, and the construction of auxiliary teaching platforms, aiming to improve the teaching quality and cultivate students' practical ability and innovative thinking.
Keywords: Teaching reform; Practical ability; Innovative thinking

1 INTRODUCTION

Object-Oriented Programming (OOP), as the foundation for constructing modern large-scale complex software systems, holds significant importance for cultivating computer science talents[1-2]. With the continuous emergence of new programming languages, frameworks, libraries, and tools, teaching content must evolve to maintain its cutting-edge and practicality. Students, as the vanguard of the era, are eager to learn the latest technologies to prepare for future challenges. This demand prompts educators to rethink the curriculum structure and integrate cutting-edge technologies into teaching. However, existing courses often face the contradiction of tight schedules and voluminous content. How to efficiently utilize limited class hours to achieve effective teaching has become an urgent issue to be resolved. This project aims to enhance teaching quality and strengthen students' practical abilities and innovative thinking. By designing teaching syllabi, writing experimental textbooks, designing auxiliary teaching platforms, and constructing teaching design cases, we explore the formation of a promotable and sustainable teaching reform model. This paper will detail the project's research contents, research achievements, features, and innovative points, as well as the application and promotion of the results, demonstrating the significant effects of teaching reform in improving students' comprehensive qualities and course teaching quality.

2 OBJECTIVES AND APPROACH

The research objectives encompass several key areas:
(1) To author an experimental textbook that meets the practical needs of teaching and emphasizes the cultivation of practical skills.
(2) To design an efficient and intelligent tool platform for assisting with or automatically grading Java code in object-oriented programming.
(3) To construct a series of innovative and exemplary teaching design cases to enhance the efficiency and quality of teaching evaluations.
(4) To promote the reform and innovation of teaching methods.
The research approach begins with key aspects of course teaching. It aims to enhance students' practical abilities and creative thinking. By integrating educational theories and computer technology from various disciplines, the project undertakes the development of experimental textbooks. It also involves designing a platform for assisting or automatically grading Java code in object-oriented programming. Additionally, it includes the study of teaching design cases. The research outcomes are applied to actual teaching processes. Through practical testing and continuous improvement, a promotable and sustainable model for teaching reform is formed.

3 RESEARCH CONTENT

3.1 Experimental Textbook

Object-Oriented Programming (OOP) is relatively straightforward when starting out, but designing systems that are clear in structure, easy to maintain, and implementable in complex systems is indeed a challenge. Additionally, the Java language involves a multitude of components, containers, and framework technologies, making the knowledge points quite extensive.
This book focuses on enabling learners to quickly grasp the syntax of the Java language and the concepts of object-oriented programming. The most efficient way to achieve this is through case-based practice. The author, drawing from years of teaching and research experience, has meticulously designed and arranged the content

organization and learning guidance in this book, addressing the characteristics of the Java language and the difficulties students encounter in their learning process.

(1) The textbook comprehensively covers the Java language knowledge system. To help learners systematically master Java, this book focuses on its core content. It extensively covers Java's knowledge points, allowing learners to continue learning important content beyond the classroom and aiding teachers in selecting relevant material for teaching and self-study. By studying this book, learners can lay a solid foundation for further study.

(2) Case-based teaching demonstrates syntax standards and programming concepts. Reading and imitating case programs and projects is an effective way to learn program design and software development. For beginners, extensive theory can be overwhelming and demotivating, but presenting theoretical knowledge through cases makes it more accessible. This book includes well-designed case studies in the preliminary knowledge section of each experimental project. These cases come with detailed descriptions, examples, code, and annotations. They explain their objectives and the problems they address. By showcasing the application of knowledge needed for experiment projects and demonstrating programming standards, methods, and ideas, learners will gradually acquire the necessary syntax, algorithms, and design techniques. Through reading, understanding, and imitating these case projects, learners will naturally develop the ability to think of solutions to experimental problems.

(3) The book integrates cutting-edge artificial intelligence knowledge into case studies to enhance practicality and novelty. "Java Programming Practice Tutorial" is a book published by the School of Computer and Information Security at Guilin University of Electronic Technology. It uses Java syntax and programming skills to implement image processing and common deep learning operations such as convolution and pooling. The book includes computer vision and artificial intelligence-related cases and experimental tasks. It combines the study of Java object-oriented programming with foundational artificial intelligence algorithms. This approach enhances the course's appeal, practicality, and cutting-edge nature.

(4) Content decoupling and a mix of difficulty levels reduce the learning curve. Each experimental project provides learners with the prerequisite knowledge and explanations needed to complete the experiment. Theoretical knowledge required for the experiment is presented concisely, accompanied by specially designed typical cases that offer various guidance and instructions for learners. This design minimizes the need for learners to consult additional theoretical materials or reference books, as the book itself meets the basic knowledge needs for the experiments. The structure of the tutorial is self-contained, so it can be used regardless of the theoretical textbook in use, without affecting students' ability to complete the experiments. The content is arranged from simple to complex, and from basic syntax to advanced applications. Each experiment lays the groundwork for the knowledge needed in subsequent ones, ensuring a gradual learning process. This approach prevents sudden difficulties and ensures that learners are never at a loss on where to start, maintaining coherence and balance throughout the content[3-4].

(5) The book bridges subsequent knowledge to enhance application development skills. As mentioned earlier, Java is widely used in Java EE system development and Android app development. Many learners lack a clear understanding of how to approach these two areas, which hinders their ability to improve in application development. To address this, the book includes four experimental projects that cover commonly used Java EE development topics such as JavaBeans, Servlets, and JSP, which can be applied to simple system development immediately after learning. It also introduces the basics of Android development. These projects allow students to quickly understand and get started with Java for Java EE and Android app development. This lays a solid foundation for enhancing their engineering application skills and boosts their interest and motivation to learn.

The use of experimental teaching materials has significantly improved students' understanding and mastery of object-oriented programming knowledge, enhanced their programming practice skills, and laid a solid foundation for their subsequent professional course studies and practical project development.

3.2 Construction of Auxiliary Teaching Platform

(1) The web-based automatic paper generation system we developed has comprehensive user management features, including roles for teachers, students, and administrators. Teachers can easily manage exams, create papers, publish tests, grade papers, and view score analysis reports. Students can take exams online, check their scores, and see detailed answers. Administrators are in charge of system maintenance and data management. The system uses various intelligent evaluation technologies to automatically and accurately grade objective questions like multiple-choice and fill-in-the-blank. For programming questions, it evaluates through code syntax analysis and comparison of runtime results, providing error prompts and suggestions for improvement[5-6].

(2) The auxiliary teaching resource system established on the Educoder platform provides students with a wealth of practical project cases and a coding practice environment. After students write their code on the platform, they can submit it for automatic grading. The platform assesses the code based on preset evaluation criteria and provides timely feedback on the grading results and suggestions for improvement. This helps students continuously enhance their code quality and programming skills.

The auxiliary teaching platform significantly enhances the efficiency and accuracy of teaching evaluations. It reduces the workload for teachers and provides timely feedback for students. This feedback helps students identify and improve their learning issues. The platform plays a crucial role in fostering students' practical programming skills and awareness of code quality.

3.3 Teaching Design Cases

Teaching design includes teaching objectives, teaching design and implementation, teaching evaluation, and appropriately introduces ideological and political content. The main content of teaching design cases includes:

(1) Knowledge objectives and moral education objectives are combined. Knowledge objectives include understanding object-oriented concepts such as classes, objects, inheritance, polymorphism, encapsulation, interfaces, exception handling, abstract classes, and generics. Students should master the basic methods of object-oriented programming and be able to apply them throughout the process of analyzing and solving complex problems in the field of computer engineering, designing basic object-oriented architectures and plans. Knowledge objectives also involve mastering practical Java platform technologies, such as commonly used utility classes, input/output streams, GUI programming, multi-threading, network programming, and database programming. Students should be able to correctly apply these technologies to solve real-world problems in computer engineering practice. Moral education objectives aim to stimulate students' national pride, patriotism, and awareness of people's livelihood during the course implementation. They aim to cultivate students' team collaboration, effective communication, and independent thinking skills. The objectives also include fostering professional qualities such as a spirit of excellence and craftsmanship, instilling good ethical behaviors like not plagiarizing or stealing, and encouraging students' self-learning and innovative consciousness.

(2) A variety of teaching methods and strategies are integrated, with appropriate introduction of ideological and political elements. The teaching design and implementation use a range of teaching methods and strategies to enhance students' learning interest, practical ability, and ideological and political quality. The teaching design includes various methods such as project-driven teaching, case teaching, and group collaborative learning. For instance, in Lesson 10 "Composition of Java Graphical User Interfaces," a problem-driven approach combined with case introduction is used, drawing an analogy between GUI composition and pasting paper cuttings for window decoration. Students are guided to consult materials and solve problems through "user registration interfaces" and "QQ login interfaces." At the same time, ideological and political elements such as "scientific literacy, exploration of the unknown, pursuit of truth, the spirit of striving for excellence, practical problem-solving ability, and the integration of learning and thinking" are introduced to cultivate students' basic scientific literacy and rigorous academic attitude. In the lecture on "Java Multithreading Development," case introduction is used, taking the 12306 ticketing system as an example to introduce concepts of processes, threads, multithreading, and shared data in multithreading. Ideological and political elements like "competition and cooperation" are also introduced, guiding students to understand the importance of resource sharing and cooperative win-win awareness. In practical projects, project introduction methods are used, where team collaboration is emphasized to foster students' team spirit. Students are encouraged to have an exploratory and innovative spirit when analyzing and solving problems. They are also guided to have a rigorous work attitude when designing schemes and debugging code. Students are taught to complete reports independently and to reject plagiarism, fostering their integrity and a truthful scientific attitude.

(3) The teaching evaluation is comprehensive and diverse. The course's evaluation system is characterized by its all-encompassing and multifaceted approach, aiming to assess students' learning outcomes and capability development from various perspectives. The evaluation system consists of three parts: regular scores, experimental scores, and final exam scores, which account for 20%, 30%, and 50% of the total grade, respectively. This proportion emphasizes both formative and summative assessment.

Regular scores focus on students' daily learning performance, including online exercises and tests, weighted at 70% and 30%. This design encourages continuous student engagement in learning activities rather than relying solely on the final exam.

The assessment of experimental scores places greater emphasis on students' practical operation skills and innovative thinking. The experimental score is composed of 50% regular experimental performance and 50% hands-on assessment, with the regular experimental performance further divided into 80% experimental completion scores and 20% experimental report scores. This detailed evaluation method more accurately reflects students' performance in the experimental process and the quality of their reports.

The final exam score assessment focuses on students' mastery of course objectives, especially their understanding and application of core concepts like classes, inheritance, and interfaces. Different levels of assessment criteria clearly define students' performance at various levels.

Additionally, the evaluation system particularly emphasizes the cultivation of integrity, requiring students to complete reports independently and refusing to plagiarize, which helps students develop a truthful scientific attitude. Overall, the course's teaching evaluation system aims to promote students' comprehensive development, focusing not only on knowledge acquisition but also on capability enhancement and character building.

The teaching design and implementation reflect a student-centered teaching philosophy, stimulating students' learning enthusiasm through diverse teaching methods, and cultivating their comprehensive qualities. It also emphasizes the cultivation of students' ideological and political qualities, enabling students to develop correct values and outlooks on life while mastering professional knowledge. This teaching design not only improves teaching effectiveness but also lays a solid foundation for students' all-around development[7-8].

4 FEATURES AND INNOVATION POINTS

The experimental textbook focuses on the combination of practice and theory, introducing a large number of actual project cases, and the cases have strong scalability and comprehensiveness, which is convenient for teachers to adjust and expand flexibly according to the actual teaching situation. The practice resource system built based on the Educoder platform provides students with a convenient practice environment, achieving seamless connection and efficient interaction in the learning process. Teaching design cases innovatively integrate various teaching methods organically, oriented by projects, centered on students, breaking the traditional teaching model where teachers lecture alone. Teaching design appropriately introduces ideological and political elements, giving full play to the main role of students and cultivating students' comprehensive qualities[9].

5 APPLICATION AND PROMOTION OF RESULTS

The research results of this project have been applied and promoted in our college. The experimental textbook has been used as the experimental textbook for the object-oriented programming course in our college, and teachers and students have feedback that the textbook content is practical and the cases are rich, which can effectively help students improve practical ability. The practice resource system built based on the Educoder platform has become an important auxiliary support for teachers in the course group, making the teaching process more vivid and efficient. Teaching design cases have been shared and promoted among course group teachers through activities such as teaching seminars and teacher training. Many teachers have applied the cases to their own classroom teaching and have achieved good teaching effects, with a significant increase in students' learning enthusiasm and academic performance[10-11].

6 TEACHING EFFECTS

Observe the participation and learning effects of students in a course number in the 2023-2024-2 semester. (1) Educoder platform, student participation and effects, as shown in Figure 1. (2) The distribution of final total scores is shown in Table 1.

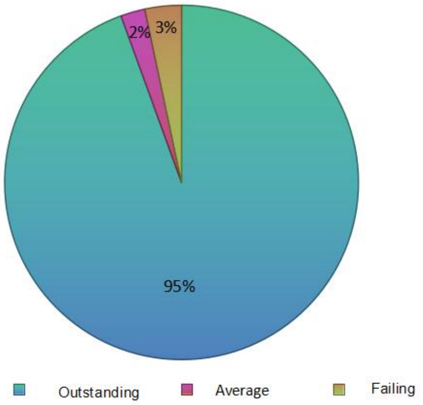

Figure 1 Average Practical Scores on the Educoder Platform (2023-2024-2 Semester)

Table 1 Distribution of Final Total Scores (2023-2024-2 Semester)

Score Range	[90, 100]	[80, 90)	[70, 80)	[60, 70)	[0, 60)
Number of People	11	38	27	10	2
Percentage	12.22%	42.22%	30.00%	11.11%	2.22%

7 CONCLUSION

The teaching and practice reform of the object-oriented programming course has yielded fruitful results. Through the research and application of experimental textbook writing, auxiliary teaching platform construction, and teaching case design, the teaching quality of the course and the comprehensive quality of students have been effectively improved, which has a high application value and promotion prospect. In the future, the project team will continue to deepen and improve the relevant results, and make greater contributions to the teaching reform of computer majors.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

FUNDING

This research is supported by Guilin University of Electronic Technology School-Level Teaching and Education Reform Project (JGB202107).

REFERENCES

[1] Jupp J C. Basic Principles of Curriculum and Instruction. Teachers College Record, 2005.

[2] Xueping Wu, Wenjie Pei. The Implementation Paths and Value Implications: Microcredentials for Teachers Offered by the Open University. Journal of Higher Education, 2023, 44(1): 99-106.

[3] Guangcai Yan. The Evolution and Development Directions of Undergraduate Major Education. Journal of Higher Education, 2024, 45(4): 58-67.

[4] Guan-ping Huang. Challenges and Path Selection in Cultivating Innovation Ability of New Engineering College Students.Theory and Practice of Education, 2024, 44(30): 9-13.

[5] Xue Li, Jiaqiong Zhang. The Connotation，Challenges and Promotion Path of the Construction of Virtual Teaching & Research Teams in Universities. Theory and Practice of Education, 2024, 44(33): 9-13.

[6] Xi He, Pingyun Luo, Hanyan Bian. Constructing High Quality Practical Curriculum System of Teacher Education. Theory and Practice of Education, 2024, 44(04): 46-51.

[7] Jie Liu, Yongqiang Zhao, Jingang Liu. Teaching Reform and Exploration of“C Programming”Course Based on OBE Concept. Theory and Practice of Education, 2022, 42(03): 61-63.

[8] M M Syeed, A Shihavuddin, M F Uddin, et al. Outcome Based Education (OBE): Defining the Process and Practice for Engineering Education. IEEE Access, 2022, 10: 119170-119192.

[9] L Romero-Untiveros, J Lara-Herrera, L Loyola-Campos. The Impact of Faculty Competencies on Engineering Education: Strategies for Continuous Improvement and Quality Assurance Within the Accreditation Process. 2023

[10] International Symposium on Accreditation of Engineering and Computing Education (ICACIT), Lima, Peru, 2023: 1-5.

[11] Jonan Phillip Donaldson, Ayana Allen-Handy. What is learning? A complex conceptual systems analysis of conceptualizations of learning. International Journal of Educational Research Open, 2023, 4: 100254.