International Journal of Education and Social Development
ISSN: 3078-9931 (Print), ISSN: 3078-2287 (Online) | Vol. 5, No. 1, 2025

Research on Java Teaching Reform Guided by Innovation and Entrepreneurship Orientation

Ming Zhao, Botao Liu

College of Computer Science, Yangtze University, Jingzhou 434023, China

Abstract: To implement the requirements for innovation and entrepreneurship education in higher education institutions during the new era and promote the deep integration of professional courses with "innovation and entrepreneurship education", this study proposes a "challenge-based teaching model oriented towards innovation and entrepreneurship" using the Java Programming course as an example. Grounded in Outcome-Based Education (OBE) and constructivist learning theory, the teaching practice follows a logical sequence of "problem identification—task design—model construction—outcome validation". Through modular restructuring of course content, a three-stage teaching framework—"foundation—enhancement—innovation"—is established. A tiered challenge task chain—"learn—do—create"—is designed, leveraging the EduCoder platform to digitally implement task-driven teaching and process-based assessment. Empirical findings from 74 undergraduate students in the Internet of Things Engineering programme demonstrate that this model significantly enhances student engagement, programming proficiency, and innovative thinking, facilitating progressive development from knowledge acquisition to competency generation. This research provides a replicable pathway for computer science curriculum reform within the context of innovation and entrepreneurship education, while also offering practical guidance for course digital transformation and competency-oriented teaching.

Keywords: Innovation and Entrepreneurship Education, Challenge-Based Task Chain, Java Programming, EduCoder, Competency-Oriented.

1. Introduction

In recent years, the Ministry of Education has successively issued multiple documents to promote the deep integration of innovation and entrepreneurship education with professional teaching in higher education institutions, providing institutional support for the reform of computer science courses [1-2]. As a core course in Computer Science and Software Engineering programmes, Java Programming plays a pivotal role in cultivating students' programming thinking and engineering practice capabilities. However, existing teaching methods still face challenges such as disconnect between content and practice, insufficient learning motivation, and monotonous assessment methods, making it difficult to effectively foster students' innovative abilities [3-4].

Addressing these challenges, this paper proposes a challenge-based Java teaching reform model for innovation and entrepreneurship education. Grounded in an industry-academia collaborative project and drawing upon Outcome-Based Education (OBE) and constructivist learning principles [5–6], the model integrates curriculum restructuring with task chain design, implemented through platform-based teaching. The research achieves an organic integration of course content and competency development through a combination of curriculum restructuring and task chain design, implemented via platform-based teaching. Concurrently, empirical validation is conducted across two dimensions: learning behaviour and learning outcomes.

2. Current State and Problem Analysis of Java Teaching

2.1. Background and Significance of Reform

Against the backdrop of innovation-driven development and the ongoing deepening of "mass entrepreneurship and innovation education", higher education curriculum reform faces new demands. The Java Programming course should transition from traditional "knowledge transmission" to a "task-driven, outcome-oriented" model [7-8]. By constructing authentic scenarios and interactive mechanisms, it should guide students to master knowledge and enhance capabilities through practice, achieving the simultaneous development of engineering practice and innovative abilities.

2.2. Current Teaching Practices and Key Challenges

Current Java teaching in higher education exhibits the following prominent issues concerning content systems, teaching models, and learning outcomes:

1) Lagging curriculum updates. The teaching content framework fails to dynamically align with industry demands, predominantly focusing on textbook knowledge points while lacking integration with real-world engineering scenarios and emerging technologies (such as multithreading, high concurrency, and API development). This hinders students' ability to develop systematic engineering cognition [9].

2) Monotonous teaching methods. Classroom instruction remains predominantly lecture-based, lacking interactive feedback mechanisms. Online resources are underutilised, resulting in low student motivation and sustained engagement [10].

3) Weak competency development. Practical components such as experiments and projects constitute an insufficient proportion of the curriculum, while assessment methods over-rely on written examinations, failing to adequately reflect students' integrated design and innovation capabilities. Some students, though proficient in language syntax, lack computational thinking and problem-solving skills.

In summary, the current Java teaching system exhibits significant shortcomings in knowledge updating, teaching methodologies, and competency orientation. It has yet to establish a closed-loop curriculum centred on Outcome-Based Education (OBE), rendering it ill-suited to meet talent development requirements under the new engineering discipline and innovation-entrepreneurship education frameworks.

2.3. Reform Objectives and Approach

Addressing these issues, this study adopts a core philosophy of"competency orientation, task-driven learning, and innovation-entrepreneurship integration" to explore a teaching reform pathway based on a task-based progression chain. The reform objectives include:

1) Reconstructing the content framework — Integrating knowledge points through modularisation to establish a three-tiered structure of "foundation—enhancement—innovation," reinforcing hierarchical progression and competency mapping between knowledge components;

2) Innovating teaching models – introducing a challenge-based task chain mechanism that decomposes course content into tiered learning tasks, stimulating student motivation through visualised task feedback and instant assessment;

3) Strengthening competency development-Through EduCoder platform, we implement project-based learning and showcase achievements, creating an integrated 'learning-doing-creating' pathway to enhance knowledge internalization and skill development.

Figure 1 outlines the research framework: guided by OBE principles → reconstructing course content → designing challenge-based task chains → platform-based instructional implementation → empirical validation. This establishes a replicable, scalable Java course reform model, providing a viable paradigm for higher education innovation and entrepreneurship programmes.

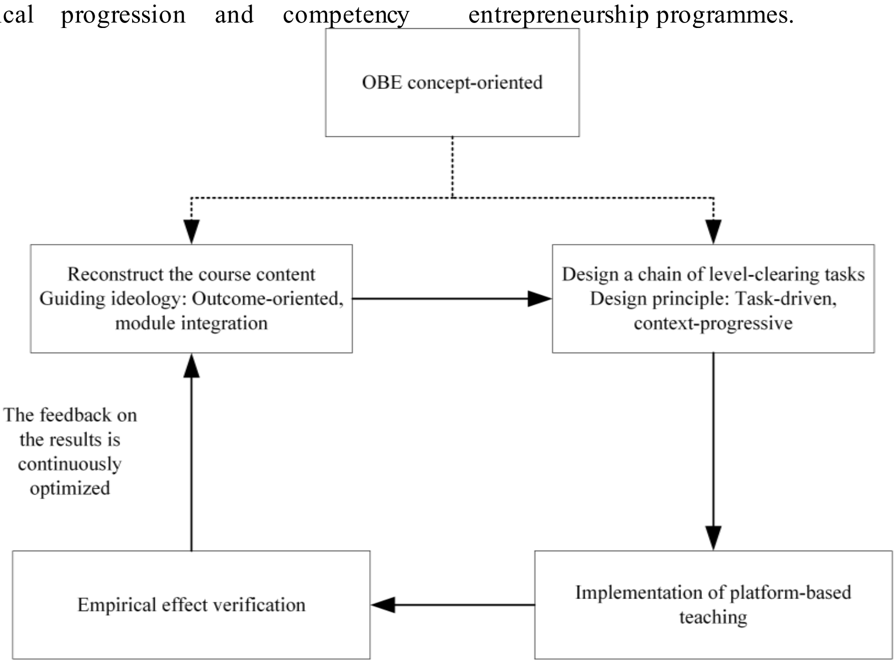

Figure 1. Overall Research Framework

Building upon this framework, Section 2 elaborates on the implementation-level "Innovation and Entrepreneurship-Oriented Challenge-Based Java Teaching Reform Strategy", encompassing course content reconstruction, pedagogical innovation, platform-based teaching implementation, and a diversified assessment system.

3. Innovation and Entrepreneurship-Oriented Java Teaching Reform Strategy Based on Challenge-Based Learning

3.1. Innovation and Entrepreneurship Education and Teaching Philosophy

"Innovation and entrepreneurship education" constitutes an educational philosophy centred on cultivating students' innovative awareness, entrepreneurial spirit, and practical capabilities. In recent years, the Ministry of Education has successively issued multiple policy documents to promote the deep integration of specialised education with innovation and entrepreneurship education in higher education institutions, providing institutional safeguards and practical space for curriculum reform. Its core objective lies in fostering the coordinated development of students' comprehensive qualities and innovative practical abilities through teaching model innovation and project-based learning. At the theoretical level, the principles of innovation and entrepreneurship education align closely with Outcome-Based Education (OBE) and constructivist learning theory. OBE emphasises learning outcomes as the guiding principle, achieving consistency between objectives, teaching content, and assessment systems through backward design of course goals [11]. Constructivism posits that knowledge should be actively constructed by learners through authentic tasks and social interaction [12].

Concurrently, the design philosophy of the challenge-based teaching model traces back to Task-Based Learning (TBL) and Gamification Learning theories. The former emphasises tiered tasks to foster knowledge application and competency development, while the latter employs levels, feedback, and incentive mechanisms to enhance learning motivation and sustained engagement. Both approaches align with the core requirements of innovation and entrepreneurship education: being student-centred and competency-oriented.

Building upon this, this study utilises the Java Programming course as a vehicle, integrating OBE principles, constructivism, and the level-based teaching model to construct an innovation and entrepreneurship teaching system centred on the student and oriented towards competency development. Through a chain of level-based tasks, project-based practice, and formative assessment, it establishes a continuous cultivation pathway from knowledge mastery to the generation of innovative capabilities.

3.2. Overall Reform Design and Implementation Approach

Guided by the aforementioned reform objectives and principles, this study further constructs a "Mentality-Driven, Challenge-Based Java Teaching Reform Strategy" at the implementation level. This strategy adheres to the overarching principles of being "student-centred, competency-oriented, task-driven, and enterprise-demand-led". It focuses on four key areas: course content restructuring, teaching model innovation, platform-based teaching implementation, and a diversified assessment system, thereby establishing a systematic reform pathway. Specifically, this includes:

1) Curriculum Content Reconstruction — Reorganising the course content framework based on OBE principles and competency-oriented requirements, forming a three-tier modular structure: "Foundation — Enhancement — Innovation";

2) Teaching Model Innovation — Introducing a tiered task chain mechanism to integrate theoretical knowledge into hierarchical assignments and project-based practice;

3) Platform-based teaching implementation — Utilising the EduCoder platform to enable learning data tracking, process visualisation, and instant feedback;

4) Diversified assessment system – Establishing a comprehensive evaluation mechanism centred on formative assessment and supplemented by summative evaluation, unifying learning behaviour, learning outcomes, and competency development.

3.3. Reconstruction of Course Content

Table 1. Java Course Content Framework Reconstruction (OBE-Oriented)

Stage	Stage Objectives	Module Number and Theme	Teaching Content	Principles of Knowledge Modularisation	Core Competencies
Stage 1: Foundations — Language Awareness and Logical Construction	Master Java's fundamental syntax and programme logic, comprehend object-oriented concepts, and develop foundational problem-solving and programme abstraction capabilities	M1: Java Syntax and Programme Logic Construction	Data types, operators, flow control, method definition and invocation	Progress through "syntax → logic → process" to gradually develop algorithmic thinking	Programming logic thinking, problem decomposition and solution capabilities
		M2: Object-Oriented Concepts and Class Design	Classes and objects, encapsulation, inheritance, polymorphism, constructors, this/super keywords	Transition from "syntactic implementation" to "object abstraction", developing awareness of class encapsulation and reusability	Ability for abstract modelling and structured programming
Stage 2: Advanced – Engineering Practices and Standard Application	Master the Java Standard Library and multithreading mechanisms, develop modular programming and exception handling capabilities, and cultivate an engineering programming mindset	M3: Java SE Core API and Collection Framework Applications	Common APIs, collection classes, generics	Organised by functional modules (data structures—algorithmic tools) to promote code reuse	Standard library application, data structure selection, and code standardisation capabilities
		M4: Exception Handling and File I/O Programming	try-catch-finally, throws mechanism, file stream and buffer stream operations	Modularised according to the "error → capture → persistent output" logic	Program robustness and file data processing capabilities
		M5: Multithreading and Concurrency Control	Thread creation, synchronisation mechanisms, thread pools, deadlock prevention	Transitioning from single-threaded to concurrent control, constructing concurrent programming models	Concurrency design and performance optimisation capabilities
		M6: Graphical User Interfaces and Event-Driven Design	JavaFX components, event listeners, MVC design pattern	From console → GUI interaction → MVC architecture, progressive design within modules	Interface interaction and event response design capabilities
Phase 3: Innovation Chapter – System Development and Creative Implementation	Comprehensive application of prior learning to complete system-level project development, demonstrating system design, innovative implementation, and collaborative capabilities	M7: Database and Network Communication Programming	JDBC database access, SQL statement operations, Socket communication, HTTP protocol	Progression from local → remote → network communication, with modular functional integration	System integration and network communication capabilities
		M8: Integrated Innovative Project Development	Team project design, module integration, interface and database interaction, multithreading optimisation, innovative feature implementation	Designing integrated task chains following the principle of "integration-expansion-innovation"	Comprehensive design capability, team collaboration, innovation implementation and outcome presentation skills

Traditional Java Programming courses typically follow a linear progression: "syntax structures → object-oriented programming → GUI and databases → integrated projects," prioritising knowledge transmission over competency development. While this framework offers relative completeness in knowledge structure, it exhibits three shortcomings: firstly, knowledge points remain fragmented, hindering students' development of systematic programming cognition; secondly, practical components often remain at a demonstrative level, lacking authentic engineering contexts; thirdly, task design is monotonous with insufficient innovation, resulting in low student motivation and sustained engagement.

To implement the dual innovation education philosophy and enhance the competency orientation and practical depth of the course, it is necessary to systematically deconstruct and restructure the teaching content within an OBE framework. Guided by the core principles of "competency-driven, task-oriented, and progressive learning", this study reconfigures the original linear knowledge structure into a three-tier modular teaching framework: "Foundation – Enhancement – Innovation". This shifts the course focus from "knowledge transmission" to "competency development", achieving a progressive progression from syntax comprehension to engineering practice, and ultimately to innovative application.

The restructured curriculum framework, as illustrated in Table 1, centres on the progressive cultivation of learning competencies. Through modular integration and competency mapping, it establishes an evolvable teaching content system aligned with the Outcome-Based Education (OBE) philosophy, providing structural support for subsequent task chain design and platform-based teaching.

3.4. Instructional Model Innovation

Figure 2. Challenge-Based Teaching Framework

Following the modular restructuring of course content, consideration must be given to effectively embedding these knowledge modules within teaching practice. To this end, this study adopts a task-driven approach, transforming modular content into a progressive, level-based task chain that establishes systematic continuity from content structure optimisation to teaching implementation.

3.4.1. Challenge-Based Teaching Framework

To achieve the organic integration of knowledge acquisition, competency development, and innovation awareness cultivation, this study constructs a closed-loop teaching framework: "Teaching Objectives – Task Design – Challenge Guide – Evaluation Feedback," as illustrated in Figure 2. Centred on task-driven learning and learner-focused pedagogy, this framework employs a tiered challenge-based task chain to systematise teaching content, gamify the learning process, and quantify learning outcomes.

Within this framework: - Instructional objectives define knowledge and competency requirements for each stage - Task design translates abstract concepts into actionable learning activities - The Challenge Guide provides essential learning guidance and error analysis - Evaluation feedback enables dynamic adjustments through formative and summative assessments Through multi-tiered challenge chains, students progressively enhance their comprehensive abilities along a learning pathway that advances "from easy to difficult, from specific points to broader contexts, and from knowledge to practical application," achieving simultaneous development of knowledge, skills, and innovation.

3.4.2. Design of Challenge Task Chains and Challenge Guides

Within the challenge task chain design, tasks are categorised into three tiers: "Learn – Do – Create":

(1) "Learn" focuses on micro-task training to reinforce programming logic and object-oriented fundamentals;

(2) "Do" centres on modular and engineering project practice, cultivating system development and problem-solving abilities;

(3) "Innovate" guides students towards independent innovation and collaborative projects, developing system integration and creative design capabilities.

The progression chain advances from foundational to advanced levels. Coupled with the EduCoder platform's instant feedback and stage-based assessments, it forms a spiral growth pathway: "knowledge acquisition → skill transfer → competency development → innovative practice". To ensure scientifically sound and effective challenge design, three core principles are adhered to: "targeting, engagement, and guidance".

(1) Targeted

Each challenge task aligns closely with course learning objectives and industry competency requirements. For instance, by simulating game world rules and character object design, students grasp abstract concepts through solving practical problems, tightly integrating learning content with real-world application.

(2) Engagement

Task scenarios incorporate real-world corporate case studies or everyday applications, fostering immersion and challenge. For instance, the "Multithreading" challenge requires students to implement simultaneous red packet grabbing by multiple users, testing their grasp of multithreading synchronisation while mirroring genuine system development demands.

(3) Guidance

Each challenge includes a "Challenge Guide" – an integrated platform resource providing task objectives, key knowledge prompts, common error analyses, and reference implementation approaches. For instance, in database connection tasks, the guide highlights typical errors (such as driver loading exceptions or SQL syntax errors) alongside solutions, enabling students to self-diagnose and refine their code design.

Through this design, students receive immediate feedback and learning support during tasks, forming a continuous learning loop where 'learning occurs through doing, and understanding through practice'.

3.5. Platform-Based Teaching Implementation

The EduCoder platform provides essential digital support for gamified learning, with its core value lying in visualising, tracking, and enabling feedback throughout the entire process of "task-driven learning implementation and outcome evaluation" [13]. The platform supports diverse programming question types (code completion, program debugging, integrated project development, etc.), featuring automated assessment and learning analytics capabilities. It records behavioural data in real time, including learning duration, task submission frequency, and error correction rates. This process-oriented information provides educators with precise learning trajectory monitoring and differentiated teaching insights, shifting instruction from "static assessment" to "dynamic diagnosis" and establishing a technological foundation for OBE-oriented competency development.

3.5.1. Implementation of the Challenge Task Chain

Leveraging EduCoder's task deployment and tiered management capabilities, this study constructed a progression-based task chain spanning the "learn-practice-create" triad. This achieves a progressive competency development pathway from knowledge mastery through project practice to innovative application (see Table 2).

Table 2. Hierarchical Classification and Implementation Methodology of the Challenge Task Chain

Tier	Task Approach	Design Objective	Competency Focus	Clearance Criteria	Assessment Method
Learning	Code completion, debugging, algorithm fragment completion, implementation of minor functionalities	Aims to establish foundational knowledge and programming proficiency, ensuring students master core syntax and coding conventions to lay the groundwork for subsequent project-based learning	① Awareness of coding conventions ② Syntax mastery ③ Debugging and problem-identification skills	Accuracy ≥ 80%, code conforming to standards, logically correct	Platform-automated assessment + real-time classroom feedback
Do	Modular Project Training	Integrating fragmented knowledge points into modular projects, emphasising engineering mindset and team collaboration, embodying the task-driven philosophy of "learning by doing"	① Module partitioning and interface design skills ② Team collaboration and task allocation capabilities ③ Code integration and testing proficiency	Functionally complete, stable operation, and clear design rationale	Platform submission + defence + tutor assessment + peer review
Open	Open Innovation Projects	Utilising open-ended integrated projects as a vehicle to cultivate students' innovative mindset, problem-solving abilities, and project management skills, embodying the core ethos of "mass entrepreneurship and innovation education".	① Problem abstraction and resolution skills ② Innovative design capabilities ③ Project management and outcome presentation abilities	Possessing innovative elements or practical value	Faculty and Industry Mentor Assessment + Outcomes Presentation

This task chain progresses sequentially from "learning" to "innovation," embodying the pedagogical logic of knowledge acquisition → competency development → innovation realisation. Through the platform's tiered task structure and instant feedback mechanism, students progressively strengthen their problem analysis, engineering implementation, and innovative thinking abilities while advancing through levels, forming a competency mapping chain grounded in the OBE philosophy.

3.5.2. Teaching Interaction and Feedback Mechanism

During instruction, teachers deploy tasks, review code, and conduct bulk assessments via the platform, while students receive immediate feedback through system prompts, teacher comments, and peer evaluations. The platform's visual learning analytics dynamically display engagement levels, task completion rates, and learning progression curves, providing data-driven grounds for personalised teacher intervention. This establishes a formative teaching loop characterised by "task-driven learning – instant feedback – dynamic optimisation", enabling students to refine learning strategies and coding abilities through continuous feedback.

In summary, the modular restructuring of teaching content, the hierarchical design of challenge-based task chains, and the technical support of the EduCoder platform collectively form the three-tiered foundation of the challenge-based teaching model. The platform's implementation not only facilitates the digital realisation of task-driven teaching but also renders the teaching process quantifiable and traceable. This provides objective data support for empirically validating reform outcomes, propelling the curriculum's transformation from knowledge-transmission to competency-generation.

3.6. Diversified Assessment System

3.6.1. Assessment Mechanism for Challenge Task Chains

To comprehensively reflect students' learning engagement and competency development throughout the challenge task chain, this study leveraged the EduCoder platform to establish a holistic assessment system centred on formative evaluation, supplemented by summative assessment. Guided by OBE principles, this system emphasises the formative approach of "assessment facilitating learning, and learning enhancing competency." It establishes quantitative metrics across two dimensions: learning behaviour and learning outcomes. These include learning duration, task completion rate, code submission frequency, programme accuracy, and innovative achievements. Through these indicators, the system dynamically tracks students' learning progress, cognitive development, and competency enhancement across the "learn-practice-create" stages, providing data support for diversified assessment.

Table 3. Design of Learning Behaviour and Outcomes Dimensions for the Challenge Task Chain Assessment Mechanism

Evaluation Dimension	Evaluation Indicator	Competency Mapping Dimension	Source of Scoring	Data Type
Learning Behaviour	Learning Duration Mission Completion Rate / Engagement Code Submissions and Error Corrections	Self-directed learning and engagement Execution and Persistence Debugging and problem-solving	EduCoder automatic recording EduCoder automatic grading EduCoder Automatic Grading	Behavioural Data Progress Data Behavioural Data
Learning Outcomes	Program Accuracy and Submission Pass Rate Module Project Functional Completeness and Code Quality Practicality and Innovation of Innovative Projects	Logical reasoning and programming standards Engineering Practice and Team Collaboration Innovative thinking and comprehensive expression	EduCoder automated grading Teacher Assessment Teacher Assessment + Peer Review	Outcome Data Outcome Data Outcome Data

This mechanism balances process and outcome, focusing not only on the accuracy and completeness of students' learning achievements but also on their behavioural characteristics and learning trajectories within tasks. This facilitates a shift from an "outcome-oriented" to a "process-oriented" approach. Assessment data will further be utilised for subsequent teaching feedback and continuous optimisation, forming a closed-loop system of "task implementation – data collection – teaching improvement".

3.6.2. Course Assessment Scheme

Within a diversified assessment framework, this study combines continuous assessment with end-of-term examinations to construct a course assessment scheme, ensuring comprehensiveness and operational feasibility in teaching evaluation. The assessment scheme specifies the composition of marks, weighting distribution, and data sources for each component (see Table 4).

Table 4. Java Course Assessment Scheme

Assessment Dimension	Grade Weighting	Assessment Method	Primary Assessment Method	Data Source
Regular Assessment	15%	Platform Challenge Task Performance (Learning + Practice)	Automatic scoring + Teacher review	EduCoder Platform
	10%	Learning Attitude or Classroom Conduct Regular assignments and practical sessions	Classroom records Teacher assessment	Face-to-face classes Teaching records
	25%	Innovation Project (Innovation)	Project Design + Presentation + Peer Review	EduCoder Platform + Offline Demonstration Defence
Final Assessment	50%	Final Examination	Written Examination	Examination Paper

This scheme adopts a "task-driven + continuous assessment" approach, emphasising students' performance and developmental trajectory within authentic tasks. Formative assessments measure learning engagement and competency development, while summative assessments evaluate integrated application and innovative outcomes. Together, they form an OBE closed-loop evaluation system centred on "competency generation", providing quantitative evidence for subsequent empirical research.

4. Reform Outcomes and Empirical Analysis

This chapter conducts empirical analysis on the implementation outcomes of the challenge-based teaching model. It utilises multi-source data—including learning logs from the EduCoder platform, results from phased tasks, and innovative achievements—based on the challenge-based task chain evaluation mechanism and course assessment scheme established in Chapter 2 (see Tables 3 and 4). The research subjects comprised 74 undergraduate students majoring in IoT Engineering. The effectiveness of the reform was systematically validated through dimensions including learning behaviour analysis, grade distribution, innovative achievements, and satisfaction feedback.

4.1. Analysis of Learning Behaviour and Task Achievement

Learning behaviour data recorded on the EduCoder platform reflects students' engagement and progress trends within the challenge-based task chain (see Table 5). The average learning duration reached 20.8 hours, with task participation and completion rates of 93.5% and 89.7% respectively, indicating high learning motivation and task persistence.

Table 5. Statistics on Learning Behaviour and Task Achievement in the Challenge Task Chain

Indicator	Average	Description
Average learning duration (hours)	20.8	Reflects learning commitment
Mission Participation Rate	93.5%	Learning Engagement Enthusiasm
Challenge task completion rate	89.7%	Learning Persistence
Average code submission frequency	8	Programming Practice Activity
Average number of bug fixes	10	Problem-solving and reflection
Average Level Clearance Rate	78%	Program Correctness and Proficiency

Data indicates that students' average pass rate in later challenge tasks was significantly higher than in earlier stages (an increase of approximately 11%), with a gradual reduction in error correction attempts. This demonstrates notable progress in programming logic, debugging, and engineering thinking. These results validate the effectiveness of the challenge task chain in stimulating learning motivation and fostering skill development.

4.2. Academic Achievement and Competency Development Outcomes

To evaluate the impact of challenge-based teaching on competency development and overall learning outcomes, students' final grades and competency mapping were analysed according to the course assessment scheme (Table 4), as presented in Table 6.

Table 6. Distribution of Course Assessment Scores and Competency Mappings

Assessment Grade	Number of Students	Percentage (%)	Corresponding Competency Level
Excellent	8	10.8	Capable of independently completing integrated projects, possessing innovative design and system implementation capabilities
Good	7	23.0	Proficient in core API applications, with modular design and team collaboration capabilities
Moderate	22	29.7	Possesses fundamental programming skills and can complete routine tasks
Pass	19	25.7	Possesses fundamental syntax and logical abilities, though engineering mindset requires further development
Fail	8	10.8	Insufficient commitment to learning or low task completion rate
Total	74	100%	—

Results indicate that approximately one-third of students achieved a 'Good' rating or higher, with an overall pass rate of 89.2%. Analysis combining platform logs with teacher assessments reveals that higher-performing students demonstrated longer learning durations and lower error rates during challenge-based tasks, reflecting a clear positive correlation between 'effort-ability-outcome'.

4.3. Enhancement of Innovation and Comprehensive Capabilities

Students formed teams based on independent projects or university-industry collaboration initiatives to participate in the university-level "Undergraduate Innovation and Entrepreneurship Training Programme", submitting 16 applications with 7 projects approved (see Table 7). Project topics spanned artificial intelligence, web system development, and visualisation applications, fully demonstrating students' knowledge integration and innovative application capabilities.

Table 7. University-Level Innovation and Entrepreneurship Project Approvals

Student Name	Project Title	Project Type	Project Level
Zheng Yuanhao Team	Meshless Reservoir Numerical Modelling System	Undergraduate Innovation and Entrepreneurship Training Programme Project	University Level
Zhu Yan Team	Research on Machine Learning-Based Boundary Condition Algorithms for Shale Gas Foam Drainage	Undergraduate Innovation and Entrepreneurship Training Programme Project	University Level
Cai Yujie Group	Innovative Integration of Chu Culture and Large Language Models	Undergraduate Innovation and Entrepreneurship Training Programme Project	University Level
Liu Zhuo Team	Web-Based Horizontal Well Staged Testing Interpretation System	Undergraduate Innovation and Entrepreneurship Training Programme Project	University Level
Zheng Huojang Team	Music Platform System Based on B/S Architecture Incorporating User Preference-Based Music Recommendations and Popularity/Traffic Analysis	Undergraduate Innovation and Entrepreneurship Training Programme Project	University Level
An Jin Team	Development of the "Dr Shao's Paediatric Massage" WeChat Mini Programme	Undergraduate Innovation and Entrepreneurship Training Programme Project	University Level
Xiao Hao Team	Research on the Construction of a Postgraduate Entrance Examination Question Bank System Platform	University Student Innovation and Entrepreneurship Training Programme	University-level

Survey feedback indicates that approximately 70% of students can independently complete fully functional system prototypes. Project completion quality significantly exceeds pre-reform levels for comparable timeframes, demonstrating that the progressive "Learn-Do-Create" model effectively fosters innovation design and outcome incubation capabilities.

4.4. Analysis of Student and Faculty Feedback

To further validate the practical outcomes of the teaching reform, the research team conducted questionnaire surveys and interviews with 74 students and 2 teaching staff following the course. Findings indicate (see Table 8): 85% of students perceived that challenge-based tasks enhanced learning engagement and perceived difficulty; 80% felt task content aligned closely with real-world development scenarios, boosting practical learning value; 88% of students indicated that the platform's instant feedback mechanism aided understanding and error correction. Over 82% of students believed that challenge-based learning promoted improvements in programming skills and problem-solving abilities, 77% felt that modular projects strengthened teamwork and engineering thinking, and 72% considered that integrated innovation projects stimulated innovative awareness and self-efficacy. Regarding teaching staff, 84% of instructors believed this teaching model significantly enhanced classroom interactivity and pedagogical feasibility.

Table 8. Teaching Satisfaction Survey Results

Survey Participants	Survey Dimension	Primary Issue	Percentage of Agreement	Key Feedback Points
Students	Learning Motivation and Experience	Challenge-based tasks enhance learning interest and sense of challenge	85%	Clear task progression, sense of achievement during challenges, and learning guidance provided by the challenge handbook
Students	Pedagogical Utility	Challenge tasks closely mirror real-world development scenarios	80%	Contextualised tasks, executable code, and strong practical focus
Students	Platform interaction and feedback	Platform provides instant assessment and feedback to aid error comprehension	88%	Automated marking and visualised feedback enable progress tracking
Students	Knowledge mastery and skill enhancement	Progression-based challenges enhance programming skills and problem-solving abilities	82%	Frequent debugging fosters logical thinking, with a marked reduction in error correction attempts
Students	Teamwork and Project Practice	Modular Projects Enhance Teamwork and Engineering Thinking	77%	Clear division of responsibilities within teams, enhanced collaborative awareness
Students	Innovative thinking and self-efficacy	Comprehensive innovation projects stimulate innovative thinking and curiosity	72%	Open-ended tasks enhance a sense of achievement and motivation for independent learning
Teachers	Teacher feedback and teaching satisfaction	Teachers believe challenge-based teaching enhances classroom interaction and pedagogical feasibility	84%	Platform support alleviates marking pressure, fostering more proactive student engagement

Survey results indicate that high proportions of student feedback concentrate on two dimensions: the platform's instant feedback and the contextualised design of tasks. This demonstrates that technology-supported, practice-oriented task design is a key factor in stimulating learning motivation and fostering competency development. A synthesis of learning behaviour data, academic performance, and questionnaire feedback reveals that the gamified teaching reform has achieved significant outcomes in enhancing learning engagement, strengthening programming and innovation capabilities, and promoting practical application. The closed-loop mechanism of "content-task-platform-assessment," constructed based on the OBE philosophy, has effectively realised the shift in teaching objectives from knowledge acquisition to competency development. This provides a viable pathway and experiential paradigm for the digitalisation and competency-oriented teaching reform of computer science courses.

5. Conclusions

This study established an innovation-oriented, gamified Java teaching model centred on the core principles of "problem-oriented, task-driven, and progressive competency development". Following the logical sequence of "problem identification—task design—model construction—outcome validation," the research systematically completed the overall optimisation of course content restructuring and teaching implementation. Results indicate that through a three-stage teaching content framework ("foundation—enhancement—innovation") and a three-tiered challenge task chain design ("learn—do—create"), students' learning initiative, problem-solving abilities, and innovative awareness significantly improved. The introduction of the EduCoder platform enabled the digital implementation and process visualisation of task-driven teaching. Empirical findings validate this model's efficacy, providing a replicable paradigm for digital and competency-oriented teaching reforms in computer science courses within the context of innovation and entrepreneurship education. Future research will explore applying this model to other programming and engineering courses, alongside integration pathways with AI teaching assistants and learning analytics systems.

Acknowledgment

Ministry of Education Collaborative Education Project

Zhao Ming (1975–), male, native of Jianli, Hubei Province, Professor, specialising in computer vision and computer security; Liu Botao (1980–), male, native of Dawu, Hubei Province, PhD, specialising in artificial intelligence and intelligent oil and gas reservoir development.

References

[1] Central Committee of the Communist Party of China, State Council. Implementation Opinions on Deepening Innovation and Entrepreneurship Education Reform in Higher Education Institutions [EB/OL]. (2015-05-04) [2015-05-14]. http://www.moe.gov.cn/jyb_xxgk/moe_1777/moe_1778/201505/t20150514_188069.html

[2] Central Committee of the Communist Party of China, State Council. Several Opinions of the GeneralOffice of the State Council on Deepening the Integration of Industry and Education [EB/OL]. (5 December 2017) [2017-12-19]. http://www.moe.gov.cn/jyb_xxgk/moe_1777/moe_1778/201505/t20150514_188069.html

[3] Xiong Fei. Research on Teaching Reform of the "Java Programming Language" Course Based on the OBE Concept [J]. Science and Technology Wind, 2023(15): 109-111

[4] Li Liping. Teaching Research on Java Programming Design in the Context of University-Enterprise Collaboration [J]. Computer Fan, 2023(16): 100-102

[5] Yang Wen, Chen Xuejiao, Zhao Lisha. Development of an Innovation and Entrepreneurship Education Curriculum System Based on OBE Principles [J]. Journal of Jilin Agricultural and Technological University, 2025, 34(04): 98-101. DOI: CNKI: SUN:TCGB.0.2025-04-022.

[6] Wang Yan, Li Zhi. An Exploration of PBL Teacher Role Transformation and Competency Requirements Based on Constructivist Theory [J]. Chinese Higher Medical Education, 2024, (02): 39-40+69. DOI: CNKI: SUN:ZOGU.0.2024-02-017.

[7] Weng Linyuan. Sustainable Development Pathways for Innovation and Entrepreneurship Education in the New Era [J]. Research and Practice in Innovation and Entrepreneurship Theory, 2023, 6(23): 105-108. DOI: CNKI: SUN:CXYL.0.2023-23-026.

[8] Yang Xuran, Wang Jingzhi. Exploring the Flipped Classroom Teaching Model Based on OBE Principles in the Context of Educational Informatisation [J]. Journal of Xingtai University, 2023, 38(02): 160-166. DOI: CNKI: SUN:XTSF.0.2023-02-024.

[9] Wang Xinhai, Li Jiandong, Li Jiena. Research and Practice of the Software Engineering Curriculum System [J]. Computer Education, 2011, (04): 70-73. DOI:10.16512/j.cnki.jsjjy.2011.04.022.

[10] Liang Lili. Reform and Practice of Blended Teaching Models for Programming Courses in the Context of Educational Digitalisation: Taking Java Programming as an Example [J]. Information and Computers,2025, 37(02): 227-229. DOI: CNKI: SUN:XXDL.0.2025-02-056.

[11] Spady W G. Outcome-Based Education: Critical Issues and Answers[M]. American Association of School Administrators, 1801 North Moore Street, Arlington, VA 22209 (Stock No. 21-00488; $18.95 plus postage), 1994.

[12] Vygotsky L S. Mind in society: The development of higher psychological processes (Vol. 86). Harvard University Press [EB/OL]. (1978)

[13] Deng Xihui. Experimental teaching practice of programming courses based on the EduCoder platform [J]. Computers and Telecommunications, 2023, (11): 45-48. DOI: 10.15966/j.cnki.dnydx.2023.11.010.