Construction and Empirical Study of a Formative Assessment System for Programming Courses in Blended Teaching Model

Lei Zheng
School of Information Science and Technology, Baotou Teachers' College, Baotou, China

Abstract: Against the backdrop of New Engineering initiatives and digital transformation in education, blended teaching has become the mainstream model for programming courses in higher education. Traditional summative assessment-the 'one exam determines a lifetime' model-often fails to accurately diagnose students' higher-order programming abilities, such as solving complex engineering problems, and cannot support the development of students' practical skills. This has also led to the common shortcoming that students 'have a good grasp of basic knowledge but weak comprehensive application abilities.' To overcome this bottleneck, this study, based on the theory of formative assessment, has established a four-dimensional integrated evaluation system encompassing 'goal decomposition, variation exploration, layered testing, and data-driven feedback.' At the same time, we conducted a two-year comparative experiment in the course 'C Programming Language,' selecting the 2024 cohort and the 2025 cohort as the control group and the experimental group, respectively, to verify its practical effects in the teaching environment. By comparing and analyzing the course goal achievement, academic performance, and learning behavior data of the two groups of students, the experimental results demonstrate that this system has significant effectiveness in enhancing students' higher-order programming skills. The data show that the achievement of the course goal 'complex problem solving' in the experimental group reached 0.72, significantly higher than the control group's 0.62, an overall increase of 16.1%. On the other hand, the proportion of students who failed to achieve this goal dropped sharply from 44.4% to 16.2%. In addition, the final grade distribution of the experimental group was also significantly optimized, with students showing a marked increase in engagement, initiative, and effective use of feedback during the learning process. This study proves that this formative assessment system can be effectively integrated with blended teaching, practically addressing students' skill gaps and genuinely implementing the teaching principle of 'assessment for learning.' This not only provides a practical case for meeting the 'continuous improvement' requirements in engineering education accreditation, but also serves as an effective reference for teaching and assessment reforms in similar programming and engineering practice courses.

Keywords: Formative Assessment; Blended Teaching; Programming Course; Course Objective Attainment; Teaching Reform

1. Introduction

With the accelerated evolution of a new round of technological revolution and industrial transformation, emerging technology industries represented by artificial intelligence and big data have placed higher demands on the cultivation of engineering talents [1]. Against this backdrop, China vigorously promotes the construction of Emerging Engineering Education (3E), aiming to cultivate interdisciplinary talents with exceptional innovative spirit and solid practical skills. For majors such as computer science, electronics, and information technology, this implies that course teaching must transcend traditional knowledge transmission and place greater emphasis on cultivating students' higher-order abilities, such as analyzing complex engineering problems, system designing, and innovative problem-solving [2]. Concurrently, the wave of digital transformation in education and post-pandemic teaching practices have made the online-offline blended learning model increasingly the norm in higher education [3]. However, the widespread practice of blended learning has also exposed deeper challenges in teaching: how to scientifically and effectively evaluate the learning process and outcomes under this model, and particularly how to construct a formative assessment system that can both track the learning process and promote capability development, has become an urgent issue for current teaching reform [4,5]. This challenge is closely linked to the "student-centered, outcome-oriented, and continuous improvement" principles advocated by Engineering Education Professional Certification. It urgently requires a new model of course assessment that runs through the entire teaching process, provides dynamic feedback, and drives teaching improvements. Therefore, exploring and constructing a course assessment system that is adapted to blended learning environments, centered on formative assessment, and precisely targets the achievement of student capabilities is not only a response to the requirements of 3E construction and professional certification but also a key path to deepening teaching reform and enhancing the quality of talent cultivation [6].

2. The Current State of Teaching in Foundational Programming Courses

By introducing blended teaching into the classroom, a new situation has been created for the reform of programming courses. However, the traditional teaching evaluation system has not been able to adapt to this new teaching model, and the existing evaluation system lacks effective support for cultivating students' higher-order abilities. This contradiction between 'teaching' and 'evaluation' is especially prominent in courses that emphasize practical skills, such as 'C Programming'. To address this, this study conducted an in-depth analysis of course goal achievement assessment reports from the past two academic years, focusing on the core issues present in the current teaching evaluation.

2.1 Significant Shortcomings in Higher-Order Abilities

The evaluation data of the 2024 class shows that the achievement rate of Course Objective 4 (analyzing and solving complex problems) is only 0.62, barely meeting the course objective, with 16 students failing to meet the standard, accounting for 44% of the total number of students. This fully exposes the students' severe lack of ability in independently analyzing complex engineering problems, as well as in code debugging and error correction, which has become the weakest link in the teaching of this course.

2.2 A Clear "Ability Gap" Phenomenon

In the achievement report of the 2025 class, the attainment rate for Course Objective 3 (ability to solve complex engineering problems) was 0.72, significantly lower than the attainment rates for the fundamental course objectives (the attainment rates for Objectives 1 and 2 were 0.88 and 0.89, respectively). Six students (about 17%) did not meet the standard. This reveals a common competency gap among students, namely a solid grasp of basic grammar but weaker comprehensive application and innovative problem-solving skills. The teaching has failed to effectively guide students to transition from lower-order cognition to higher-order thinking.

2.3 Relatively Singular Assessment Methods

The traditional evaluation model still excessively emphasizes summative exams such as mid-term and final exams, lacking continuous tracking and effective feedback on students' key abilities in analysis, design, and debugging during the learning process. Most of the existing assessment content is limited to memorization and simple application of knowledge points, and cannot comprehensively and accurately reflect the dynamic growth of students' programming thinking and practical abilities.

2.4 Insufficient Teaching Interaction

Both teaching objective assessment reports show that although students highly recognize the teacher's teaching attitude, they consistently hope to "increase classroom interaction" and "strengthen guidance for case study discussions." From the students' perspective, this precisely confirms the limitations of the existing teaching model in stimulating deep learning and promoting knowledge transfer.

In summary, this study aims to address a core issue: how to construct a course evaluation system that is deeply integrated into the entire blended learning process. We will take formative assessment as the core driving force, by diagnosing learning behaviors and providing timely feedback, it fills the students' ability gaps to improve their programming practice and innovation abilities when facing complex engineering problems.

3. Construction of the Formative Assessment System

In response to the many issues mentioned in the previous chapter, this study plans to use formative assessment as the core theory to develop a course evaluation system integrated into the blended teaching process. The purpose of this evaluation system is to personalize the promotion of students' skill development. The core concept of the system is to facilitate the transformation of the evaluation function—from being a "judge" that assesses learning outcomes to being a "coach" that promotes learning throughout the entire learning process. By systematically collecting learning evidence, providing timely feedback, and making dynamic adjustments, this system can effectively enhance students' programming practice abilities.

3.1 Core Philosophy: Shifting from "Assessment of Learning" to "Assessment for Learning"

The theoretical foundation of this system originates from formative assessment theory. Unlike summative assessment, which focuses on certification and grading, the core of formative assessment is 'collecting, analyzing, and using evidence to improve teaching and learning.' It is not a series of isolated tests or assignments but a complete set of planned and continuous teaching processes. The key is to use the evidence generated by assessment activities to accurately bridge the gap between students' current learning status and the expected learning goals. In the context of engineering education accreditation, this means that teaching assessment directly serves the achievement of learning outcome indicators and runs throughout the entire teaching process to effectively implement the concept of "continuous improvement." Therefore, this system abandons the rough approach of simply weighting quiz and assignment scores into the final grade and instead deeply explores the formative function of assessment. Specifically, each assessment activity is anchored to specific stage-based competency goals, with the primary purpose of providing precise feedback to teachers and students — that is, to clearly indicate "the current level of the learner" and "how to progress towards the goal," rather than merely giving a cold, meaningless score.

3.2 System Overall Model: A Four-Dimensional Closed-Loop System

Based on the above concept, and combined with the characteristics of programming courses that emphasize logical thinking and practical operation, this study has constructed the formative evaluation system model shown in Figure 1.This model takes course objectives as the logical starting point and the blended learning environment as the implementation field. It operates through four interconnected core dimensions to form a "goal-evidence-feedback-adjustment" continuous improvement closed loop [10,11].

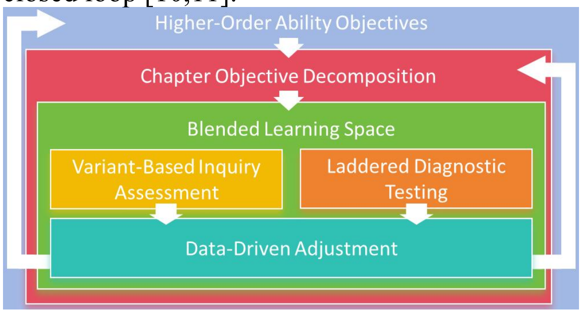

Figure 1. Formative Assessment System Model for Programming Courses in Blended Learning Mode

This model reflects the characteristic of formative assessment as an integrated process. It not only includes the design of assessment tasks, but also the improvement measures for teachers and students, combining assessment with teaching through this series of measures.

3.3 Core Strategies

3.3.1 Strategy one: assessment dimension decomposition based on course objectives

Based on Bloom's Taxonomy of Educational Objectives, The overall objectives of the C language course are broken down into a series of step-by-step, phased sub-goals. For example, Divide the learning objectives of the 'Function Application' chapter into four parts: (1) understanding function definitions, (2) being able to correctly call functions, (3) being able to design functions to solve subproblems, and (4) combining multiple functions to solve complex problems. Each sub-goal has clear, observable, and measurable success criteria, such as code correctness, module independence, and algorithm clarity. This information will be clearly communicated to students through the learning platform before the course begins to ensure that both teachers and students have a shared understanding of the chapter's objectives.

3.3.2 Strategy two: "case variant" inquiry-based assessment

This strategy is the core of teaching evaluation in the classroom, aiming to address the common dilemma among students of 'understanding the lesson in class but not being able to write the code themselves.' The specific operation follows the principle of 'teach one example, practice one variation,' and includes the following aspects.

Case Explanation: The teacher explains typical program cases related to the current skill sub-goal in class, such as calculating factorials using recursion or the bubble sort algorithm.

Variation Questions: After explaining the cases, a set of carefully designed variation questions is posted on the learning platform, requiring students to complete the implementations in class.

Types of variations include:

(1) Parameter Adjustment: changing the scale of data or data types.

(2) Logic Transformation: slightly altering the logic of the problem, such as changing ascending sorting to descending sorting, or modifying the loop termination conditions, etc.

(3) Functionality Extension: For example, adding functions to find the maximum value or calculate the average after sorting.

Multi-Dimensional Immediate Feedback: The instructor collects evidence of student learning through methods like classroom observation, selecting students to demonstrate, and organizing group discussions. Feedback not only focuses on whether the final code is correct but also emphasizes evaluating the clarity of students' thinking, the effectiveness of their debugging process, code standardization, and the innovation of their solutions, thereby diagnosing problems and guiding deeper thinking in real-time.

3.3.3 Strategy three: laddered diagnostic testing in the blended environment

This strategy builds a closed loop for assessment in online self-directed learning. After each chapter's study concludes, a "laddered" diagnostic test is released via the learning platform. Test questions are designed with progressive difficulty:

Basic Level: Tests the direct application of core concepts and syntax (e.g., multiple-choice, code completion).

Variant Level: Applies logical or functional transformations to classic cases (similar to in-class variants, but more comprehensive).

Comprehensive Level: Designs small programming projects that comprehensively apply knowledge from the current and previous chapters to solve practical problems.

The platform automatically grades objective questions and provides run-result comparison and common error hints for programming questions. After the test, the system automatically generates a learning progress report, visualizing class-wide common weaknesses and individual ability distribution.

3.3.4 Strategy four: data-driven feedback and teaching adjustment

This teaching strategy ensures that evaluation evidence and information are transformed into actual teaching actions. Teachers regularly analyze student learning progress reports (from classroom observations, various exercises, and periodic tests). Based on the collected data, actions are adjusted on three levels:

Class level: Adjust the teaching focus and pace for the next stage, and design additional explanations and exercises targeting common mistakes.

Group level: Organize students with similar problems and form study groups for collaborative learning.

Individual Level: Push personalized supplementary learning resources, extension exercises, or offer one-on-one online Q&A appointments to students who have not met the standards through the platform.

4. Implementation and Effectiveness

To validate the effectiveness of the aforementioned theoretical framework, this study conducted a two-year teaching practice in the C Programming course at the School of Information Science and Technology, Baotou Teachers' College. The Class of 2024 Computer ISEC class (n=36) was designated as the control group, maintaining the traditional summative assessment model of "lecture + midterm/final exams." The Class of 2025 Computer Class 2 (n=37) was designated as the experimental group, fully implementing the four-dimensional integrated formative assessment system of "Objective Decomposition–Variant Exploration–Laddered Testing–Data-driven Feedback" constructed in this paper. By comparing the differences between the two groups in terms of course objective achievement rates, academic grade distribution, learning process behaviors, and subjective feedback, the effectiveness of the reform was empirically tested.

4.1 Implementation Process: Differential Design for Control and Experimental Groups

Implementation for the Control Group (Class of 2024): Employed conventional blended learning. The online platform was primarily used for distributing course materials and announcements. Offline classes were mainly teacher-led lectures supplemented by limited in-class exercises. Course assessment consisted mainly of regular assignments (20%), midterm exam (20%), and final exam (60%). The regular grade was primarily based on homework submission and attendance, lacking fine-grained diagnosis and feedback on the learning process.

Implementation for the Experimental Group (Class of 2025): Teaching and assessment were strictly conducted according to the system described in Section 3. Its core lies in deeply embedding assessment into the entire "pre-class–in-class–post-class" workflow:

Pre-class (Online): Students study micro-lectures via the platform and complete pre-class self-tests. The system automatically generates preliminary learning data.

In-class (Offline Core): After the instructor precisely explains a core case, an immediate "case variant" inquiry session is organized. For example, after explaining "array sorting," students are required to immediately complete programming and debugging for variant tasks such as "descending sort" or "counting the number of swaps during the sorting process." The instructor collects learning evidence and provides targeted guidance through observation, questioning, and peer review.

Chapter Feedback Mechanism: After completing a chapter, students participate in the platform's 'stepped' chapter challenges. This testing system includes three levels: 'basic consolidation, variant application, and comprehensive expansion,' focusing on code completion, program debugging, and small-scale programming abilities. The system automatically scores the tests and generates multidimensional class and individual learning analysis reports, identifying common issues among students and analyzing individual knowledge weaknesses.

Continuous Feedback and Dynamic Teaching Adjustment: Teachers conduct in-depth analysis of the learning data reports for each chapter, dynamically optimizing subsequent teaching strategies based on this, and targeting students who have not met the standards with personalized tutoring resources. For example, if students are observed to have common errors in the 'pointers and function parameter passing' module, teachers promptly add specialized theoretical explanations and reinforcement exercises to achieve precise and immediate instructional interventions.

4.2 Implementation Effectiveness: Comparative Analysis Based on Multi-Source Data

In order to comprehensively assess the effectiveness of teaching reform, this study compared the performance of the control group and the experimental group across four dimensions, including: achievement of course objectives, academic performance, process orientation, and course evaluation.

(1) Fundamental Improvement in Course Objective Achievement Rates

Course goal achievement rate is one of the most important indicators for measuring teaching effectiveness. Compared with the control group, the experimental group performed better in achieving all course goals, especially showing the most significant improvement in higher-order ability goals, as shown in Table 1.

Table 1. Comparison of Course Objective Achievement Rates between Control and Experimental Groups

Course Objective	Control Group Achievement Rate	Experimental Group Achievement Rate
Objective 1	0.80	0.88
Objective 2	0.77	0.89
Objective 3	0.62	0.72

In the assessment targeting higher-order skills (Goals 3/4), the experimental group outperformed the control group, with an overall increase of 16.1%. These data suggest that by breaking down complex problems into a progressive sequence of tasks and providing process-oriented feedback, this teaching and assessment system can effectively guide students in transitioning from 'understanding knowledge' to 'integrated application' skills.

Another point worth noting is that the system performed outstandingly in reducing course failure rates. According to course data, in the target dimensions representing advanced skills, the number of students in the control group who did not meet the standards was 16 (44.4%), whereas in the experimental group, this number decreased to 6 (16.2%). These data demonstrate that the system not only achieved an overall improvement in teaching quality, but also, through precise diagnostics and personalized intervention mechanisms, significantly reduced the risk of students encountering difficulties in programming courses failing, thereby achieving more balanced educational outcomes.

(2) Comprehensive optimization of academic grade distribution

In the end-of-term course evaluation, the grade distribution of the experimental group was more balanced and reasonable. Compared to the 'high failure rate' and grade polarization mentioned in the control group's report, the experimental group's data more closely followed a normal distribution. The proportion of students achieving good or higher grades reached 56.76%, while the failure rate dropped to 2.70%. This distribution indicates that the formative assessment system not only promoted overall improvement in student performance but also significantly reduced the risk of failing.

(3) Positive transformation in learning process behaviors

Furthermore, through analyzing learning process data, the reasons for the improvement in academic performance were examined. The data shows that students in the experimental group exhibited higher frequency and deeper engagement in learning under the blended teaching model. Specifically, this includes the following aspects.

Classroom exercise participation: The experimental group achieved a completion rate of over 92% in the classroom 'variant exercises' session, significantly higher than the control group's classroom exercise participation rate of about 65%.

Online self-directed learning: The total number of resource visits on the teaching platform by the experimental group reached 165,086, with both per capita visit frequency and online learning duration significantly higher than those of the control group. This indicates that these students not only completed the assigned tasks but also actively used the platform resources for independent review, consolidation, and extended learning.

Use of test feedback: Over 80% of the experimental group students actively consulted and attempted to correct the error analysis generated by the system after completing the 'stage tests,' effectively forming a closed-loop learning habit of 'testing—diagnosis—improvement.'

The above behavioral data strongly demonstrate that the system, by designing cognitive-challenging variation tasks and combining them with precise real-time feedback mechanisms, effectively stimulates students' intrinsic learning motivation and promotes a substantive shift in learning behavior from 'passive reception' to 'active inquiry.'

(4) Feedback from students' subjective evaluations

As one of the core dimensions for assessing the effectiveness of teaching reforms, student feedback provides important evidence. Anonymous evaluation data from the experimental group show that students' satisfaction in aspects such as classroom interaction atmosphere, the relevance of cases to practical work, and the timeliness of learning feedback is significantly higher than that of the control group.

Many students mentioned in their comments that "understanding the programming logic through the variant problems" and "the stage tests were like passing levels, allowing us to check our learning progress at any time." This stands in sharp contrast to the appeals in the control group report "to increase classroom interaction and case teaching," confirming that the new assessment model better meets students' learning expectations and improves their learning experience.

5. Conclusion

Aiming at the realistic dilemmas in programming course assessment under blended learning, such as "prominent shortcomings in higher-order abilities" and "lack of process feedback," this study, guided by formative assessment theory, constructed and implemented a four-dimensional integrated course assessment system: "Objective Decomposition–Variant Exploration–Laddered Testing–Data-driven Feedback". Through a two-year controlled experiment in the C Programming course, this study draws the following main conclusions and discusses its value and limitations.

The constructed formative assessment system can effectively enhance students' higher-order programming abilities. Empirical data shows that the experimental group (Class of 2025), which fully implemented this system, achieved a 0.72 achievement rate for the higher-order course objective representing complex engineering problem-solving ability, a 16.1% improvement compared to the control group (Class of 2024, achievement rate 0.62) which used the traditional assessment model.

On the other hand, the proportion of failing students also significantly decreased, dropping from 44.4% in the control group to 16.2% in the experimental group. This indicates that in this study, by breaking down complex competency goals into manageable phased tasks and integrating them into a continuous cycle of assessment and feedback for improvement, it not only effectively helps students master basic grammar but also enhances their ability to solve comprehensive problems.

By integrating an assessment system throughout the entire teaching process, overall learning outcomes were effectively optimized, and students' academic performance improved. Data analysis revealed that the experimental group not only performed well in achieving core competency goals but also had a more favorable overall grade distribution: the proportion of students with 'good' grades increased significantly, while the failure rate remained low. Moreover, process data indicated that, compared with the control group, students in the experimental group showed higher classroom participation, were more active in online self-directed learning, and made better use of feedback information.

The above results confirm that the system, relying on the 'variant exploration' task design and timely 'data-driven' feedback mechanism, has completed the transformation of the course evaluation function, shifting from an external 'measurement tool' to an active 'learning engine' for students. This transformation effectively stimulates students' learning initiative and meta-cognitive awareness, truly achieving a deep shift in learning from 'exam-oriented' to 'ability development-oriented'.

Based on the background of engineering education accreditation, this study established an observable, operable, and verifiable model for continuous course improvement. The entire research process comprehensively implemented the core concepts of 'Outcome-Based Education (OBE)' and 'continuous improvement'; it diagnosed teaching pain points through authoritative evidence such as course objective achievement reports; subsequently, it designed and implemented reforms in the evaluation system; and verified the effects of course improvement through a new round of achievement reports and multi-source data. This evidence-based iterative optimization approach not only supports professional accreditation requirements but also provides a clear and concrete practical reference for promoting spiral improvements in course quality.

Acknowledgments

This paper is supported by Baotou Teachers' College University-Level Quality Engineering Teaching Reform Project (No. BSJG25Y06).

References

[1] Cao, Fei. The Frontier Trends and References in Formative Assessment. Journal of Educational Science of Hunan Normal University, 2024, 23(2): 87-96.

[2] Zheng Shizhong, Liu Cong. Construction of School-Based Research and Training System for the Development of Teachers' Digital Teaching Ability under the "Learner-Centered" Guidance. Journal of Jilin Provincial Institute of Education, 2026, 42(544): 35-45.

[3] Feng, Xiaoying, Wang, Ruixue, & Wu, Yijun. A Review of Domestic and International Blended Teaching Research: Based on an Analytical Framework for Blended Teaching. Journal of Distance Education, 2018, 36(3): 13-24.

[4] Li, Haidong, & Wu, Hao. Research on a Whole-Process-Based Blended Teaching Quality Evaluation System: Taking a National-Level Online-Offline Blended First-Class Course as an Example. China University Teaching, 2021(5): 65-71.

[5] Yao, Chunxu, Zhu, Yamei. Review and Prospect of Research on Blended Teaching in Domestic Higher Education.Journal of Higher Education, 2024(12): 91-95.

[6] Liu, Yangbing, et al. Analysis and Application of Formative Assessment in Concrete Structure Design Principle Course Based on Blended Teaching Mode. Journal of Higher Education, 2026(2): 42-44.

[7] Li, Zhiyi, Li, Qingqing. On Discrimination of Process-based Assessment and Formative Assessment: From Engineering Education Accreditation Perspective. Research in Higher Education of Engineering, 2022(5): 6-11.

[8] Sun, Jing, & Liu, Xin. Teaching Practice of Processual Evaluation, Formative

Evaluation, and Summative Evaluation. Research in Higher Education of Engineering, 2024(4): 94-100.

[9] Zhou, Lin, & Zhou, Wenye. Formative Assessment: Cultivating Students' Self-Regulated Learning Ability. Testing and Evaluation, 2020(2): 53-57.

[10] Black P, Wiliam D. Inside the black box: Raising standards through classroom assessment. Phi Delta Kappan, 1998, 80(2): 139-148.

[11] Garrison D R, Anderson T, Archer W. Critical inquiry in a text-based environment: Computer conferencing in higher education. The internet and higher education, 1999, 2(2-3): 87-105.