ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202605_9(5).0001

Research on the Evaluation System of Course Objective Attainment for Computer Science Programs under the OBE Framework

Qian Wang[1, a]

[1] School of Computing and Artificial Intelligence, Shandong University of Finance and Economics, Jinan 250014, China

[a] qianwang@sdufe.edu.cn

Abstract

Outcome-Based Education (OBE) has become an important guiding philosophy in higher education reform, especially in engineering and computer science disciplines. The OBE approach focuses on learning outcomes and emphasizes whether students have achieved the expected abilities after completing the learning process. However, in many computer science courses, evaluation methods still rely heavily on final examination results, which cannot fully reflect students' comprehensive abilities such as programming skills, practical problem-solving capability, and collaborative learning capacity. To address these limitations, this study proposes a systematic evaluation framework for course objective attainment under the OBE concept. First, the hierarchical relationship among program objectives, graduation requirements, and course objectives is analyzed to establish a clear mapping structure. Second, a multi-dimensional evaluation indicator system is constructed by integrating process assessment, final examination performance, and practical project evaluation. Third, a quantitative calculation method for course objective attainment is introduced to improve the objectivity and accuracy of teaching evaluation. A case study based on the course Data Structures is conducted to validate the effectiveness of the proposed evaluation framework. The results demonstrate that the evaluation system can effectively measure students' learning outcomes and provide reliable feedback for teaching improvement. The proposed approach contributes to improving the scientific evaluation of learning outcomes and promoting continuous improvement in computer science education.

Keywords

OBE; computer science education; course objectives; evaluation system.

1. Introduction

In recent decades, the rapid development of information technology and digital industries has significantly increased the demand for high-quality computer science professionals. Universities are expected to cultivate graduates who not only possess solid theoretical knowledge but also demonstrate strong practical abilities and innovative thinking. Under this background, traditional teaching models that emphasize knowledge transmission are gradually being replaced by competency-oriented educational approaches.

Outcome-Based Education (OBE) has gradually become an influential educational philosophy in higher education reform. The central idea of OBE is to evaluate educational effectiveness based on students' learning outcomes rather than merely focusing on teaching activities [1]. By clearly defining expected competencies, educators can design curriculum structures, teaching strategies, and evaluation methods that better support students' ability development.

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202605_9(5).0001

In engineering and computer science education, the OBE concept has been widely adopted in international accreditation systems. For example, the ABET accreditation criteria emphasize outcome evaluation and continuous improvement as key principles of engineering education[2]. Through systematic assessment of learning outcomes, universities can ensure that educational programs remain aligned with industry needs.

However, in many computer science courses, evaluation methods still rely heavily on traditional examinations. Such evaluation approaches mainly assess students' theoretical knowledge and may not fully reflect other important competencies, including programming ability, teamwork, and practical problem-solving skills[3].

Therefore, establishing a comprehensive evaluation framework for course objective attainment has become an important task in teaching reform. A well-designed evaluation system can provide reliable feedback for improving teaching quality and promoting continuous curriculum improvement.

2. OBE-Based Teaching Objectives in Computer Science Education

2.1. Concept of Outcome-Based Education

Outcome-Based Education (OBE) is an educational philosophy that focuses on students' learning achievements after completing the educational process. Compared with traditional teaching models that mainly emphasize knowledge delivery, OBE stresses the measurable abilities that students should obtain at the end of learning activities[1]. It shifts educational attention from what teachers teach to what students are actually able to demonstrate after learning[2]. These outcomes often include professional knowledge, practical skills, problem-solving abilities, communication competence, teamwork awareness, and lifelong learning capacity. In engineering and computer education, OBE has been widely adopted because it provides a clear framework for aligning talent cultivation with industrial demands[3].

One of the most important characteristics of OBE is its student-centered orientation. In this framework, teaching activities are organized according to the competencies students are expected to acquire. Teachers first determine learning outcomes and then design appropriate teaching strategies and evaluation methods to support these outcomes[4]. Students are encouraged to actively participate in the learning process rather than passively receiving information. This learner-centered approach is particularly important in adult education because adult learners usually have diverse professional backgrounds, practical experiences, and personalized learning needs.

Another essential feature of OBE is backward curriculum design. Instead of starting with teaching content, educators begin by identifying expected learning outcomes. Curriculum structure, teaching methods, and assessment strategies are then developed based on these outcomes to ensure that teaching activities are aligned with educational objectives[5]. This design approach helps improve curriculum coherence and ensures that every learning activity contributes to talent development goals.

Furthermore, OBE emphasizes continuous improvement in teaching. Through systematic evaluation of learning outcomes and course objective attainment, educators can identify weaknesses in teaching activities and adjust curriculum design accordingly. This feedback-based improvement process helps universities maintain educational quality and adapt to changing technological environments [6]. In the era of large language models, OBE provides an effective framework for redesigning adult computer education to better cultivate human-AI collaborative competence.

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202605_9(5).0001

2.2. Mapping Relationship between Program Objectives and Course Objectives

Under the OBE framework, educational objectives are usually structured hierarchically. This structure generally includes program objectives, graduation requirements, and course objectives. The hierarchical design helps ensure that educational activities at different levels remain logically connected and collectively contribute to talent cultivation goals. It also enables universities to establish a systematic mechanism for curriculum planning, teaching implementation, and quality assurance.

Program objectives describe the competencies that graduates are expected to possess several years after completing their academic programs. These objectives are usually formulated according to industry development trends, national educational policies, and professional accreditation standards. They emphasize long-term career development and social adaptability. For example, graduates of computer-related programs may be expected to become professionals who can solve complex engineering problems, adapt to technological changes, demonstrate ethical responsibility, and contribute to digital transformation in various industries. In adult higher education, program objectives may also place greater emphasis on practical problem-solving ability and career advancement because many adult learners return to education for professional development purposes.

Graduation requirements provide more detailed descriptions of the abilities students should acquire before graduation. They serve as a bridge between broad program objectives and specific teaching activities. In computer science programs, these competencies typically include problem analysis, algorithm design, system implementation, teamwork, communication ability, engineering ethics, and lifelong learning capability[2]. With the rapid development of artificial intelligence technologies, graduation requirements may also include new competencies such as human-AI collaboration, digital literacy, and the ability to critically evaluate AI-generated outputs.

Course objectives represent the learning goals of individual courses. Each course contributes to specific graduation requirements through its teaching content and assessment activities. For instance, a programming course may focus on algorithmic thinking and coding ability, while a software engineering course may emphasize teamwork and project management skills. Therefore, establishing a clear mapping relationship between course objectives and graduation requirements is essential for ensuring that curriculum design supports the achievement of program-level educational objectives [4].

In addition, this hierarchical objective structure helps universities evaluate whether students have achieved expected learning outcomes through measurable indicators. It also provides an important basis for curriculum adjustment and continuous improvement. In the era of large language models, such a structured framework becomes increasingly important because educational institutions must continuously update learning objectives to align with rapidly changing technological and industrial environments.

To illustrate this hierarchical relationship, Figure 1 presents the structural relationship among program objectives, graduation requirements, and course objectives under the OBE framework.

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202605_9(5).0001

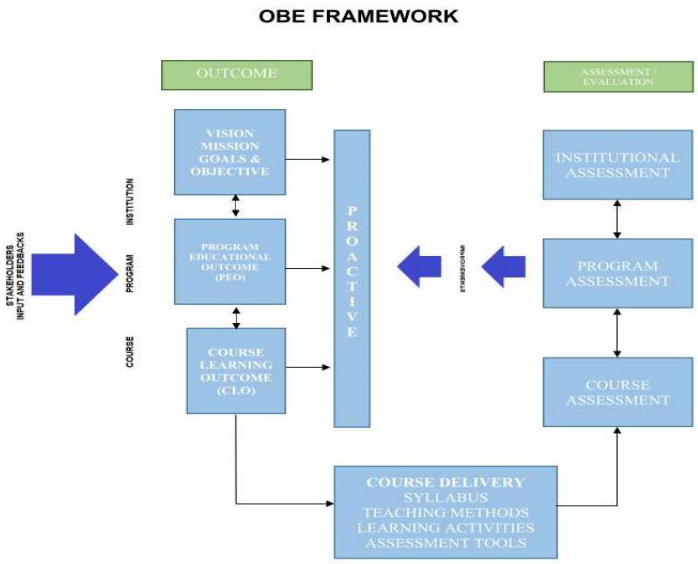

Figure 1. Structure of OBE Course Objectives

3. Evaluation System of Course Objective Attainment

3.1. Evaluation Indicators

In the OBE framework, evaluating course objective attainment should involve multiple assessment indicators rather than relying solely on final examination results. In many traditional courses, final examinations often account for the majority of the course grade. However, such evaluation methods cannot fully reflect students' comprehensive learning outcomes[7].

To better evaluate students' learning performance, multiple assessment indicators should be integrated into the evaluation system. In computer science courses, assessment activities typically include process assessment, final examinations, and practical projects.

Process assessment reflects students' continuous learning engagement throughout the semester. Final examinations mainly evaluate students' understanding of theoretical concepts. Practical projects focus on assessing students' ability to apply knowledge to real-world problems, which is particularly important in computer science education[8].

By integrating these evaluation components, educators can obtain a more comprehensive understanding of students' learning outcomes.

In practical teaching, relying on a single evaluation method may lead to an incomplete understanding of students' learning outcomes. Therefore, when constructing an evaluation system for course objective attainment, it is necessary to consider both knowledge mastery and ability development. In computer science courses, students' learning performance is often reflected not only through theoretical examinations but also through programming practice, project implementation, and collaborative learning activities. These learning activities provide valuable information for evaluating whether students have achieved the expected course objectives.

Process assessment plays a particularly important role in this context. Compared with traditional examination-based evaluation, process assessment focuses more on students' learning behavior throughout the semester. For example, homework assignments can reflect students' understanding of course content, while classroom participation and discussions can reveal students' engagement and initiative in learning. In programming-related courses,

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202605_9(5).0001

laboratory experiments and coding tasks can further demonstrate students' practical ability to apply theoretical knowledge to real-world problems.

Another important component of the evaluation system is practical project assessment. In many computer science courses, students are required to complete programming projects or experimental tasks. These activities allow students to integrate multiple knowledge points and apply them to complex problem-solving scenarios. By evaluating students' performance in such tasks, teachers can gain a more comprehensive understanding of students' analytical ability, algorithm design capability, and programming skills.

Furthermore, combining different evaluation indicators helps reduce the bias caused by any single assessment method. For instance, a student who performs poorly in a final examination may still demonstrate strong practical ability through programming assignments or project work. Therefore, integrating process evaluation, theoretical examinations, and practical projects can provide a more balanced and reliable assessment of course objective attainment. This multi-dimensional evaluation approach is consistent with the core principle of OBE, which emphasizes evaluating students' comprehensive learning outcomes rather than relying solely on examination scores.

3.2. Calculation Method of Course Objective Attainment

In order to measure the achievement level of each course objective, a quantitative evaluation method can be introduced. Suppose that a course contains several course objectives, and each objective corresponds to specific assessment tasks. The attainment level of the *ith* see Table 1.

$$A_i = \frac{\sum_{j=1}^{n} S_{ij}}{T_i}$$

Where A_i represents the attainment level of the *i-th* course objective,

S_{ij} denotes the score obtained by students in the *j-th* assessment related to the objective, and

T_i represents the total possible score for that objective.

Table 1. Evaluation indicators for course objective attainment

Evaluation Component	Weight
Process assessment	30%
Final examination	40%
Practical project	30%

3.3. Evaluation Process

A systematic evaluation process is essential for ensuring the effectiveness of the course objective attainment evaluation system. The evaluation procedure usually includes defining course objectives, establishing mapping relationships with graduation requirements, designing diversified assessment activities, collecting evaluation data, calculating attainment levels, and analyzing evaluation results.

To illustrate this process clearly, Figure 2 presents the workflow of the course objective attainment evaluation process.

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202605_9(5).0001

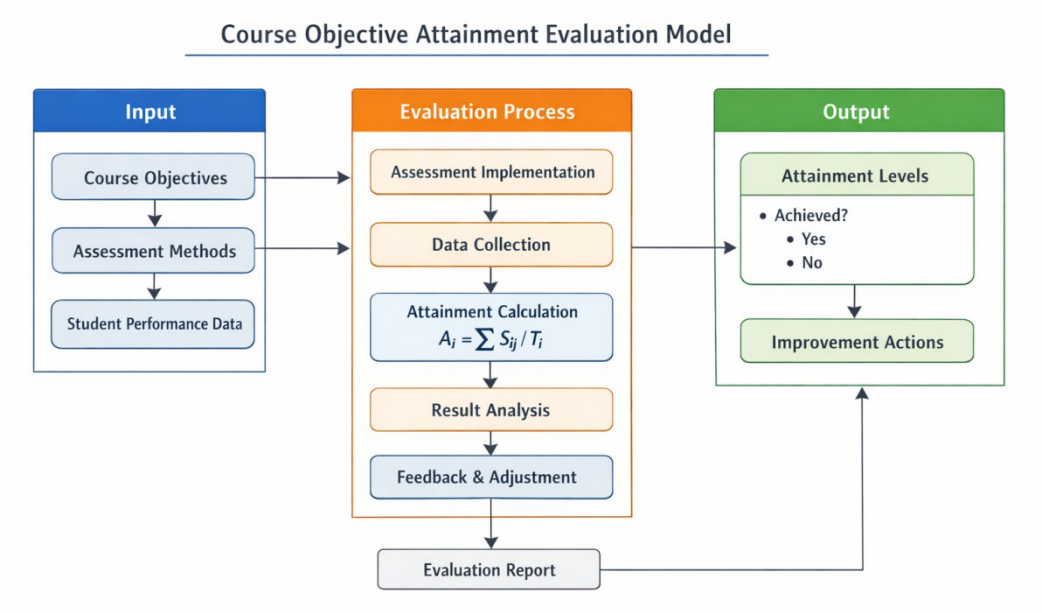

Figure 2. Evaluation Process for Course Objective Attainment

The evaluation process forms a closed-loop structure. Evaluation results not only reflect students' learning outcomes but also provide important feedback for improving teaching strategies and curriculum design. This continuous improvement mechanism is one of the core principles of OBE-based education.

4. Case Study

To verify the effectiveness of the proposed evaluation system, a case study was conducted in the course Data Structures, which is a core course in computer science programs. The course mainly focuses on fundamental data organization methods, algorithm design, and problem-solving skills. It plays an important role in cultivating students' programming ability and computational thinking.

According to the OBE framework, several course objectives were defined for the Data Structures course. These objectives were designed to support the achievement of relevant graduation requirements in computer science education. Table 2 presents the main course objectives of the selected course.

Table 2. Course objectives of Data Structures

Course Objective	Description
Objective 1	Understand basic concepts and principles of data structures
Objective 2	Design and implement algorithms to solve practical problems
Objective 3	Analyze algorithm efficiency and optimize program performance

Through this calculation, teachers can quantitatively evaluate the degree to which each course objective has been achieved. If the attainment level exceeds a predetermined threshold, the course objective is considered to be achieved. Otherwise, corresponding teaching improvements may be required. This quantitative approach reduces subjective judgment in

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202605_9(5).0001

teaching evaluation and provides more objective evidence for curriculum adjustment. It also helps departments monitor teaching quality across different semesters and supports long-term continuous improvement in educational management.

Based on the proposed evaluation system, multiple assessment methods were used to measure students' learning outcomes. These assessment components included homework assignments, programming experiments, and final examinations. The collected assessment data were used to calculate the attainment level of each course objective.

Table 3. Course objective attainment results

Course Objective	Attainment Level
Objective 1	0.84
Objective 2	0.79
Objective 3	0.87

As shown in Table 3, the results indicate that the attainment levels of all course objectives exceed the predefined threshold of 0.70, which means that the expected learning outcomes have generally been achieved. However, the attainment level of Objective 2 is slightly lower than the others, suggesting that further improvements may be needed in teaching algorithm design and practical programming tasks.

Through this case study, the proposed evaluation system demonstrates its effectiveness in measuring course objective attainment and identifying areas that require improvement. The evaluation results also provide valuable feedback for teachers to adjust teaching strategies and improve curriculum design.

5. Continuous Improvement Mechanism

One of the key principles of Outcome-Based Education is continuous improvement. After evaluating the attainment level of course objectives, the evaluation results should be used to improve teaching activities and curriculum design[9]. Through systematic analysis of evaluation data, teachers can identify weaknesses in the teaching process and make appropriate adjustments to enhance learning effectiveness[10]. This reflects the closed-loop quality assurance mechanism emphasized in OBE, in which teaching implementation, outcome evaluation, feedback collection, and curriculum improvement form an interconnected cycle.

In computer science education, continuous improvement can be implemented from several aspects. First, teaching content can be updated according to evaluation results and industry development trends. For example, if students show insufficient achievement in programming practice or algorithm design, additional exercises, laboratory activities, or practical training tasks may be introduced into the course. Emerging technologies such as artificial intelligence, cloud computing, and cybersecurity can also be incorporated into course content to ensure curriculum relevance.

Second, teaching methods can be improved to enhance student engagement and learning outcomes. Teachers may adopt diversified teaching approaches, such as project-based learning, case analysis, flipped classrooms, and collaborative learning, to strengthen students' practical abilities and problem-solving skills. In adult education settings, flexible online learning resources may also be introduced to accommodate learners with different schedules.

Third, the assessment system can be optimized to better reflect students' comprehensive abilities. By adjusting the weight of different assessment components and introducing more practical evaluation methods, the evaluation system can provide a more accurate measurement of course objective attainment.

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202605_9(5).0001

Through the implementation of a continuous improvement mechanism, evaluation results can effectively support the optimization of curriculum design and teaching strategies, thereby improving the overall quality and adaptability of computer science education.

6. Conclusion

This paper proposes an evaluation system for course objective attainment in computer science programs under the Outcome-Based Education framework. By analyzing the relationship among program objectives, graduation requirements, and course objectives, a structured evaluation framework is established. A multi-dimensional evaluation indicator system is constructed by integrating process assessment, final examinations, and practical tasks.

Furthermore, a quantitative method for calculating course objective attainment is introduced, which enables teachers to measure the achievement level of course objectives more accurately. A case study based on the Data Structures course demonstrates that the proposed evaluation system can effectively evaluate learning outcomes and provide useful feedback for improving teaching quality.

The results indicate that the proposed approach helps enhance the scientific evaluation of course objectives and supports the continuous improvement of teaching activities. More importantly, it strengthens the alignment between curriculum design and talent cultivation objectives, ensuring that teaching activities better meet the competency requirements of computer science education.

In addition, the proposed framework provides practical guidance for universities that are currently promoting engineering education accreditation and curriculum reform. It can help administrators establish standardized evaluation procedures and improve the transparency of teaching quality assessment.

In the future, further research may focus on applying the evaluation system to more computer science courses, comparing evaluation results across different institutions, and exploring the integration of digital technologies such as learning analytics and artificial intelligence to support more efficient and intelligent teaching evaluation.

Acknowledgements

This research was supported by the Shandong Natural Science Foundation Project (ZR2023MF039).

References

[1] W. G. Spady. Outcome-Based Education: Critical Issues and Answers, American Association of School Administrators, Arlington, VA (1994).

[2] ABET Engineering Accreditation Commission. Criteria for Accrediting Engineering Programs, ABET, Baltimore, MD (2019).

[3] R. M. Harden. Outcome-based education: the future is today, Medical Teacher, vol. 29 (2007), 625–629.

[4] R. M. Felder, R. Brent. Designing and teaching courses to satisfy the ABET engineering criteria, Journal of Engineering Education, vol. 92 (2003), 7–25.

[5] J. Biggs, C. Tang. Teaching for Quality Learning at University, Open University Press, Maidenhead (2011).

[6] M. Prince, R. M. Felder. Inductive teaching and learning methods: Definitions, comparisons, and research bases, Journal of Engineering Education, vol. 95 (2006), 123–138.

ISSN: 2637-6067 DOI: 10.6918/IJOSSER.202605_9(5).0001

[7] D. Boud, N. Falchikov. Aligning assessment with long-term learning, Assessment & Evaluation in Higher Education, vol. 31 (2006), 399–413.

[8] G. Wiggins, J. McTighe. Understanding by Design, Association for Supervision and Curriculum Development, Alexandria, VA (2005).

[9] L. W. Anderson, D. R. Krathwohl. A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Longman, New York (2001).

[10] J. M. Spector. Foundations of Educational Technology, Educational Technology Research and Development, vol. 63 (2015), 1–15.