

# Reforming Python Programming Teaching Based on Objective Problem-Oriented Approach from Constructivism Perspective



Yue Cao\*, Jixin Liu

School of Automation, Guangdong University of Petrochemical Technology, Maoming 525000, China

**Abstract:** In response to the prevalent challenges in Python programming education, including the intricate nature of knowledge points, diminished student engagement, and suboptimal programming proficiency, this research introduces a goal-oriented, problem-based pedagogical framework underpinned by constructivist principles. The study meticulously designs targeted problems that are intricately aligned with the curriculum, deployed at strategic intervals before, during, and after class sessions. These problems are crafted to bolster students' initiative, foster enthusiasm, and amplify their sense of challenge. The pedagogical strategy is further enriched by integrating focused discussions on these problems, in-depth analysis of pivotal knowledge points, and interactive extended dialogues, thereby rendering the teaching process more dynamic and captivating for learners. This methodology is engineered to ignite students' interest in the subject matter and to galvanize their drive to tackle problems, with the ultimate objective of nurturing and enhancing their capacity to autonomously devise programs that address practical issues. The efficacy of this approach has been substantiated through two cycles of teaching practice, yielding outcomes that are deemed to be highly satisfactory. The results underscore the potential of this model to revolutionize Python programming instruction by making it more student-centered, interactive, and aligned with real-world problem-solving, thereby equipping students with the skills necessary to navigate the complexities of programming in a rapidly evolving digital landscape.

**Keywords:** Constructivism; Objective-Problem Based; Teaching Model; MOOC Platform

**DOI:** [10.57237/j.edu.2025.01.002](https://doi.org/10.57237/j.edu.2025.01.002)

## 1 Introduction

The Python language has garnered increasing attention and application due to its unique advantages of being "easy to learn, free and open-source, and rich in library resources." Particularly, its application in the field of artificial intelligence is notably leading. The most common machine learning frameworks, such as TensorFlow, PyTorch, and Baidu's PaddlePaddle, all provide excellent support for Python. In the annual programming language rankings by IEEE Spectrum and the TIOBE index (the world's most authoritative programming language ranking), Python has consistently held the top position for

many years [1].

As a relatively "older" new engineering discipline, the Intelligent Science and Technology major aims to cultivate high-quality specialized technical talents with scientific thinking abilities, including computational thinking, and the ability to design computational solutions and implement systems based on computational principles. These talents are competitive in research, development, deployment, and application related to computational systems. One of the core courses in this major is "Python Programming." The mastery of this course by students

\*Corresponding author: Yue Cao, [caoyue2000@gdupt.edu.cn](mailto:caoyue2000@gdupt.edu.cn)

majoring in Intelligent Science and Technology significantly impacts the learning outcomes of subsequent advanced intelligent courses. The target audience for this course is second-year students who have already established a solid foundation in computer science and mathematics. Before taking this course, students have already completed the "C Programming" course. Therefore, they already possess a certain level of learning ability and programming language foundation, allowing them to naturally transfer previously acquired knowledge and way of thinking to this course. However, based on previous teaching outcomes, students' mastery and application of Python after completing the course are relatively low, even affecting their performance in subsequent courses [2].

The flexibility and ease of use of Python are a "double-edged sword." While it is easy to get started, mastering it and developing computational thinking to solve practical problems remains challenging. The traditional teacher-centered teaching model is not conducive to students taking initiative or developing computational thinking. From the perspective of constructivist theory, adopting a goal-oriented problem-based teaching model that emphasizes the practical application of Python can enhance students' initiative and challenge. By proposing goal-oriented questions before, during, and after class that are closely tied to the course content, students are encouraged and guided to solve these problems, thereby increasing their learning initiative and engagement. In teaching practice, interactive activities such as goal-oriented discussions, in-depth analysis of key points, and interactive extended discussions are used to enhance the interactivity of teaching activities and students' sense of participation, stimulating their interest in learning and cultivating their ability to solve problems through independent programming.

## 2 Constructivist Learning Theory

From the late 1970s to the early 1980s, constructivism gradually became an important theoretical school in cognitive psychology. Jerome Bruner, one of the main representatives of constructivism, emphasized the process by which people actively encode, interpret, and reconstruct information to build knowledge and understanding. The true proposer of constructivist theory was Jean Piaget, the founder of the "Geneva School" and a renowned philosopher. His idea of "bidirectional construction between sub-

ject and object" laid the theoretical foundation for individual constructivism. Although constructivist theory is rich in content, its core can be summarized in one sentence: it centers on the student, emphasizing active exploration, discovery, and the construction of meaning from the knowledge learned (rather than merely transferring knowledge from the teacher's mind to the student's notebook, as in traditional teaching). Student-centered teaching emphasizes "learning," while teacher-centered teaching emphasizes "teaching." This is the fundamental divergence between the two educational philosophies, leading to the development of two opposing learning theories, teaching theories, and instructional design theories. Since the learning environment required by constructivism is strongly supported by the latest information technology achievements, constructivist theory has increasingly been integrated into the teaching practices of educators, becoming a guiding ideology for deepening teaching reforms in schools both domestically and internationally [3-6].

## 3 Goal-Oriented Problem-Based Teaching Model

The goal-oriented problem-based teaching model sets relevant questions based on the professional training objectives, course teaching objectives, and knowledge point objectives. Through methods such as pre-class preparation, in-class teaching, and group discussions, teachers and students work together to guide the completion of the course learning according to the set questions, achieving the course objectives. To ensure the effectiveness of the goal-oriented problem-based teaching method, it is essential to carefully design the goal-oriented questions [3]. Generally, the goal-oriented problem-based teaching method sets five types of questions based on the teaching content: basic questions, key questions, difficult questions, practical questions, and extended questions. These five types of questions cover the entire learning process of students. The specific settings of these five types of questions are as follows: basic questions are usually simple questions targeting the concepts of knowledge points, which students solve during pre-class preparation; key questions are important knowledge points that require detailed explanation by the teacher during class; difficult questions are for students to think about, with guidance from the teacher; practical questions are discussed and

answered by students in groups; and extended questions are for students to study independently after class. The key to the goal-oriented problem-based teaching method is the teacher's careful design of the five types of questions based on the teaching objectives, ensuring the smooth conduct of teaching activities and the achievement of training objectives [7].

The goal-oriented problem-based teaching model is an excellent practice of constructivist theory. Constructivist learning theory advocates a student-centered approach, emphasizing students' active exploration, discovery, and construction of meaning from the knowledge learned. In other words, its core task is to shift from how to teach to how students learn. The goal-oriented problem-based teaching model perfectly implements the core content of constructivist learning theory by setting five types of questions and combining them with classroom discussions [8]. Basic questions require students to learn independently, with the more challenging parts set as key and difficult questions. In class, students actively think, with guidance and explanation from the teacher. Practical questions are discussed and answered by students in groups, and extended questions are completed by students through self-study after class. The five types of questions stimulate students' initiative, encouraging them to explore and learn. In this process, students gain a sense of achievement, which in turn makes them more proactive, forming a positive feedback loop. In terms of the active construction of meaning from the knowledge learned, the goal-oriented problem-based teaching model emphasizes the teacher's thorough understanding and grasp of the professional training objectives, course teaching objectives, and knowledge point objectives. This ensures that students' active construction is not aimless but guided by clear objectives. It can be said that the goal-oriented problem-based teaching model is an excellent practice of constructivist learning theory [9-15].

## 4 Teaching Practice of Python Programming

After thoroughly analyzing the relationship between the aforementioned constructivist learning theory and the goal-oriented problem-based teaching model, a teaching practice of the goal-oriented problem-based teaching model from the perspective of constructivism was conducted for the Python programming course in the Intelli-

gent Science and Technology major. In this process, the core of constructivist learning theory was closely adhered to, serving as the theoretical guide for the reform of the goal-oriented problem-based teaching model. The specific process is as follows:

- 1) Before class, students are assigned to read and study specified content and watch teaching videos. Basic questions are provided for them to answer, aiming to achieve the basic knowledge point objectives.
- 2) During class, based on the basic questions, students are guided, encouraged to think, and explore actively. Key and difficult questions are proposed for detailed explanation and discussion, aiming to achieve the key and difficult knowledge point objectives. At the same time, based on the knowledge gained from solving basic and key questions, practical questions are introduced for in-class discussion and solution, further deepening the understanding of knowledge points and giving students a sense of achievement, thereby stimulating their learning initiative.
- 3) After class, extended questions are assigned for students to think about and solve. The setting of extended questions should be cautious, not too difficult, and should require some effort based on the practical questions. This improves students' ability to solve practical problems independently and further stimulates their interest in learning.

Here, the loop structure in the program flow control of the "Python Programming" course is taken as an example for detailed explanation.

- 1) Before class, through WeChat or QQ groups, specified reading chapters and learning videos are pushed, and students are required to answer basic questions:
  - 1) What is a loop structure? How many ways does Python have to implement a loop structure? How to implement them? Write programs to calculate the sum of 1 to 100, the sum of even numbers, and the sum of odd numbers; calculate the value of the natural logarithm base e until the absolute value of the last term is less than  $10^{-6}$ . Students independently answer these basic questions and complete simple programs using two types of loop control.
- 2) During class, after randomly asking and commenting on students' answers, students are guided to recall the similarities and differences between Python and C language loop control. This naturally leads to the role of the break and continue statements. Then, key and difficult questions are discussed: nested loops

and the performance analysis of nested loops. The traversal of multi-dimensional data structures is used to introduce nested loops, and actual code is shown to demonstrate the results, further explaining the uses of nested loops (such as traversing multi-dimensional data structures, generating permutations and combinations, printing specific patterns, search and matching problems, and their combination with nested conditional judgments). It is also noted that nested loops are widely used in subsequent intelligent-related courses, and their performance needs to be analyzed. Several programs are provided for students to read, explain their functions, and analyze the execution results. An online quiz is conducted through the Chaoxing Learning Platform or MOOC platform, with multiple online test questions assigned, and the results are automatically generated on-site. The teacher provides on-site evaluation and encouragement based on the test results.

3) After class, students are assigned materials for further independent thinking and learning, such as the

combination of nested loops and conditional statements, the combination of nested loops and list comprehensions, and the analysis of the execution effects of adding break and continue in nested loops. These are assigned as homework for future course tutoring sessions.

Among these, practical questions can be assigned before class and encouraged to be completed by students, with explanations and evaluations during class. Alternatively, they can be proposed during class, with students participating in discussions and writing code on the spot to complete the programs. In teaching practice, the latter method can more immediately engage the classroom atmosphere, and students unknowingly actively participate in solving practical problems.

According to the teaching design and the tools used in online teaching, the assessment methods for students' learning status are shown in Table 1. On the management interface of the Chaoxing Learning Platform or MOOC platform, these behavioral data statistics can be easily obtained for teachers to manage and supervise students' learning.

Table 1 Assessment of Students' Learning Status

No.	Assessment Content	Percentage
1	Completion of all course video viewings	5%
2	Completion of all online exercises	10%
3	Completion of all programming assignments	10%
4	Final Exam Results	60%
5	Attendance and Check-ins	5%
6	Online Quizzes	10%
7	Submission of Extended Problem Reports	Extra Credit

## 5 Teaching Reform Effectiveness and Student Feedback

According to the online teaching evaluations from the second semester of the 2022-2023 academic year at our university, students rated the goal-oriented problem-based teaching model above 95 points. Detailed assessment metrics revealed that approximately 85% of students believed this teaching model could stimulate their learning initiative and interest, while nearly 90% felt it effectively improved their programming skills and overall learning outcomes. However, about 85% of students also noted that it consumed too much of their extracurricular time, particularly regarding extended problems. Most students

relied on teacher explanations rather than completing these tasks independently. To address this, future efforts will include providing concise lecture materials and other solutions.

## 6 Conclusion

This paper discusses constructivist theory as the theoretical foundation for the goal-oriented problem-based teaching model and positions the model as a practical application of constructivist theory. It explores the cognitive psychological basis of the goal-oriented problem-based teaching model at a theoretical level. Additionally, it introduces the teaching practice process of the goal-oriented problem-based teaching model in the Python programming course, using actual course content as an example for detailed explanation. The practical effects of the

teaching reform and existing issues are also presented. As a core foundational course in the Intelligent Science and Technology major, Python programming demands both breadth and depth in its content. Future efforts will focus on addressing students' lack of autonomy in solving extended problems.

## References

- [1] Wei Dongmei, Wang Xiuhua, Wang Ying, et al. Construction and Teaching Practice of General Python Programming Course [J]. *Computer Education*, 2019(2): 69-73.
- [2] Song Tian, Huang Tianyu, Li Xin. Python Language: An Ideal Choice for the Reform of Programming Course Teaching [J]. *China University Teaching*, 2016(2): 42-46.
- [3] Ji Quanpeng, Yan Haisheng. Teaching Reform and Practice of Python Programming Course - Based on Project-Based Teaching Model [J]. *Journal of Southwest China Normal University (Natural Science Edition)*, 2021, 46(11): 90-95.
- [4] Lu Di, Li Su, Liu Shengzhi, et al. Application and Practice of Project-Based Teaching in the "Python Programming Language" Course [J]. *Technology Wind*, 2024, (27): 120-122.
- [5] Duan Xilong, Wu Zhihao. Online and Offline Hybrid Teaching of Object-Oriented Programming Course Based on BOPPPS [J]. *Journal of Higher Education*, 2023, 9(23), 104-107.
- [6] Lu Ye, Zhang Xiaoli, Shi Lingwei, et al. Reform of Programming Course from the Perspective of Hybrid Teaching Space. *Computer Education*, 2023(10): 96-101.
- [7] Ding Lei. Research on the Reform of Bilingual Teaching in Advanced Language Programming Guided by Constructivism. *Computer Knowledge and Technology*, 2012, 17(18): 107-108.
- [8] Yu Xiaoping, Li Zhigang, Gao Pan, et al. Teaching Practice of Java Programming Based on UbD Concept and Constructivist Theory. *Computer Education*, 2023(02), 54-57+63.
- [9] Qin Dawei, Luo Tianyu, Chen Qi. Exploration of a Dynamic Progressive Teaching Model Based on Goal-Problem Orientation. *Journal of Higher Education*, 2024, 10(26): 98-101.
- [10] Wang Hui, Liang Bingbing, Liu Hongxian. Teaching Reform and Practice of the "Introduction to Petrochemical Processes" Course Based on Goal-Problem Orientation. *The Guide of Science & Education*, 2024(19): 120-122.
- [11] Lai Jinhui. Research on Blended Teaching Mode of Programming Fundamentals in MOOC Environment [J]. *Experimental Science and Technology*, 2017(5): 77-81.
- [12] Zhao Zhifeng. Research on Ideological and Political Teaching Reform Based on Goal-Oriented Problem-Based Approach [J]. *Heilongjiang Science*, 2022, 13(9): 158-161.
- [13] Lai Jinhui. Exploration of Ideological and Political Teaching in Programming Courses Based on Goal-Oriented Problem-Based Teaching Model [J]. *Computer Education*, 2023(05): 50-54.
- [14] Wang Li, Shan Shufeng, Sun Jin, et al. Blended Teaching Reform and Practice of Petrochemical Technology Based on Goal-Oriented Problem-Based Approach [J]. *Chemical Enterprise Management*, 2023(01): 55-59.
- [15] Lai Jinhui. Research and Practice of Goal-Oriented Problem-Based Discussion Teaching Model in Python Programming [J]. *Education of Industry and Information Technology*, 2023 (02): 13-16.