Research and Practice of Teaching Reform for "Python Programming" under the New Engineering Background

Luyin Ma[1, a]

[1] Xi'an Technological University, Xi'an, Shaanxi, China

[a] 2867388058@qq.com

Abstract. With the rapid development of information technology, programming skills have become a fundamental requirement in modern society. As a foundational course for computer science, Python programming plays a critical role in cultivating students' coding skills and innovative thinking. However, traditional teaching models and content no longer meet current educational demands, making curriculum reform imperative. This paper analyzes the current state of Python programming courses and proposes a teaching reform plan. The reform focuses on updating course content by integrating the latest Python technologies and applications, such as data analysis and machine learning; adopting innovative teaching methods, including project-driven teaching and flipped classrooms, to enhance students' practical abilities and teamwork; and strengthening practical components through phased project tasks to improve problem-solving skills. Additionally, the course evaluation system has been optimized to use diversified assessment methods, emphasizing process-based performance and project outcomes. After the reform, students demonstrated significant improvements in programming skills, overall quality, class engagement, and interest in learning, as well as enhanced practical skills and innovative thinking. Teachers also improved their teaching capabilities through training and resource integration. In the future, Python programming courses will continue to optimize teaching content and methods, explore innovative teaching models and application scenarios, and better align with technological development needs to cultivate high-quality talent with innovation and practical abilities.

Keywords: Python programming；Teaching reform; project-driven； Flipped classroom； Innovative teaching models

1. Introduction

1.1 Research Background. Python is a free, open-source, cross-platform, high-level dynamic programming language known for its extensibility, readability, and simplicity. It has gradually become the preferred programming language in emerging industries such as artificial intelligence, big data, and cloud computing. In March 2018, the Ministry of Education included Python in the National Computer Rank Examination, and "Python Programming" became a subject for the Level II computer examination. Since 2021, Python has surpassed traditional compiled languages like C and Java to become the most popular language in the TIOBE programming language rankings. Currently, Python is widely used in web development [1], financial analysis [2], big data mining [3], web scraping[4], and many other fields. The demand for Python talent in the job market far exceeds supply under the new engineering background. Python programming courses not only provide non-computer science students with tools for data analysis in subsequent professional fields but also lay the foundation for computer science students to learn new technologies such as machine learning [5] and deep learning[6]. Therefore, universities must emphasize the innovative construction and practice of Python courses in computer science education, guiding students to leverage Python as a tool to solve various complex problems in learning and life.

To adapt to industry development needs and the trend of educational informatization, reforming the teaching of Python programming courses is essential. Updating course content, innovating teaching methods, and strengthening practical components can effectively improve students' programming skills, innovative thinking, and practical application abilities. Such reform not only nurtures more competitive computer science talent but also provides valuable experience for

building curriculum systems in higher education, promoting high-quality development in higher education.

1.2 Research Purpose and Significance. Python, as a programming language that is both simple to learn and powerful in functionality, has been widely adopted worldwide in recent years. Whether in data analysis, artificial intelligence, web development, or the Internet of Things, Python plays an indispensable role, making it one of the most important technologies in the IT industry. With Python's growing influence, incorporating it into programming course systems in universities has become a trend. However, traditional programming courses often fail to fully meet the demands of current technological developments and the need for cultivating interdisciplinary talent.

The construction of new engineering disciplines is a key focus of current higher education reform, emphasizing interdisciplinary integration and future-oriented innovation. Python, as a widely applicable cross-disciplinary programming language, holds unique advantages in new engineering education. Through teaching reform and exploring Python-based teaching models, this research aims to provide valuable experiences for constructing new engineering curricula.

In terms of teaching models, introducing flipped classrooms, project-based learning (PBL), and blended teaching methods can build more flexible and efficient teaching systems that meet students' personalized learning needs [7]. Integrating real-world cases, interdisciplinary projects, and industry challenges into course teaching not only enhances students' ability to tackle complex engineering problems but also promotes the depth and sustainable development of new engineering education. Furthermore, optimizing the evaluation system to combine process-based and outcome-based assessments can comprehensively reflect students' overall abilities and innovative achievements, laying a solid foundation for cultivating high-quality engineering talent for the future .

This research aims to provide guidance for computer science curriculum reform in universities and contribute practical experiences to the development of new engineering education, offering both theoretical value and practical significance.

2. Current Situation Analysis of Python Programming Course

2.1 Survey on Student Learning. A survey revealed several common issues students face during learning that impact their effectiveness and programming skills development. Firstly, understanding challenges such as Python syntax rules, data structures, and algorithm logic are frequently mentioned. For example, some students struggle with foundational concepts like object-oriented programming and exception handling, which slow their progress. Secondly, a lack of practical ability is another major problem [8]. Although practical components are included in the course, these tasks are often limited to simple coding exercises, lacking training in real-world problem scenarios and project development. As a result, students lack systematic design and project management skills, unable to effectively apply theoretical knowledge to practice. These findings highlight the need for reform to improve students' interest in Python programming courses, address learning challenges, and strengthen practical skills.

2.2 Current State of Teaching Equipment, Materials, and Extracurricular Resources. At present, while most Python programming courses are equipped with basic computing equipment and programming software, there are still shortcomings. For example, some equipment experiences frequent failures due to delayed maintenance and updates, disrupting classes [9].

Regarding teaching materials, although many Python textbooks are available, most are too basic and lack in-depth coverage of Python's applications in fields like data science, artificial intelligence, and web development. These materials often focus on foundational knowledge and algorithms, failing to keep pace with technological advancements [10]. Additionally, many schools lack comprehensive resource repositories, making it difficult for students to access the latest practice tutorials and open-source projects.

2.3 Utilization of Online Learning Platforms. The proliferation of information technology has made online learning platforms vital supplementary tools for teaching Python programming. Many universities have started using platforms like MOOCs and Bilibili to provide extracurricular

learning resources. However, some challenges remain.

While these platforms offer abundant learning materials, many students do not fully utilize them for independent learning. Furthermore, the content on these platforms often does not align well with classroom teaching, leaving students confused and unsupported when encountering difficulties. Finally, the interactive features and practical components of some platforms are underdeveloped, limiting students' ability to receive real-time feedback during exercises [11].

In summary, while teaching equipment, materials, and extracurricular resources support Python programming education to some extent, there is still room for improvement. The integration of online learning platforms into teaching requires optimization to better align with classroom instruction and enhance students' learning outcomes and autonomy.

2.4 Limitations of Traditional Teaching Content and Methods. Currently, most universities still adopt traditional teaching content and methods for Python programming courses, which have significant shortcomings. Firstly, the course content is overly basic, often limited to Python syntax rules, basic data structures, and simple algorithm explanations, lacking extensions to Python's applications in data analysis, artificial intelligence, web development, and other fields [12]. This limits students' interest in the course and the depth of their learning.

Secondly, the teaching method is singular, primarily focused on theoretical lectures and classroom demonstrations, with students passively absorbing knowledge and lacking opportunities for independent exploration and interactive participation. This approach leads to a monotonous classroom atmosphere, hindering students' enthusiasm and practical skills development. Furthermore, the design of practical components is often overly simplistic, with tasks disconnected from real-world applications, making it difficult for students to genuinely master programming skills and leaving them feeling unprepared when facing complex problems.

2.5 Gaps Between Students' Skill Needs and Course Design. The modern workplace demands diverse abilities from computer science students, including a solid programming foundation, interdisciplinary application capabilities, teamwork, and innovative problem-solving skills. However, the current design of Python courses fails to fully meet these demands.

Firstly, course content updates lag behind, lacking coverage of industry-leading technologies and real-world application scenarios, leaving students unaware of Python's critical role in fields like data science and artificial intelligence. Secondly, practical tasks are often small-scale, single-function programs, lacking comprehensive projects that could effectively cultivate students' system design abilities [13]. Additionally, assessment methods focus primarily on exams, neglecting to evaluate students' practical skills, teamwork, and learning process, thereby limiting their comprehensive development.

3. The specific measures of Python programming teaching reform

The following paper elaborates the specific measures of python programming teaching reform from four aspects: teaching content, teaching method, teaching practice, and evaluation system, as shown in Fig. 1.

3.1 Introduce the latest Python technology and application cases. In order for Python programming courses to keep up with the pace of technological developments, course content needs to introduce the latest Python technologies and application areas in a timely manner. In recent years, data analysis, machine learning and artificial intelligence have become important areas of Python application, and these technologies have a wide range of applications in academic research, industrial production, financial analysis and other industries. The curriculum should increase the content of these technologies, so that students not only learn the basics of programming, but also understand and master how to solve practical problems with Python.

For example, data analysis cases can be introduced to teach students how to use libraries such as Pandas and NumPy in Python for data cleaning, processing and visualization to help students understand the value behind the data [14]. Machine learning content can introduce basic methods of building models using tools such as Scikit-learn and TensorFlow to stimulate students' interest in AI technology. Through the design of simple application cases, students can gradually master these

popular technologies while learning Python, and improve their programming ability and practical ability.

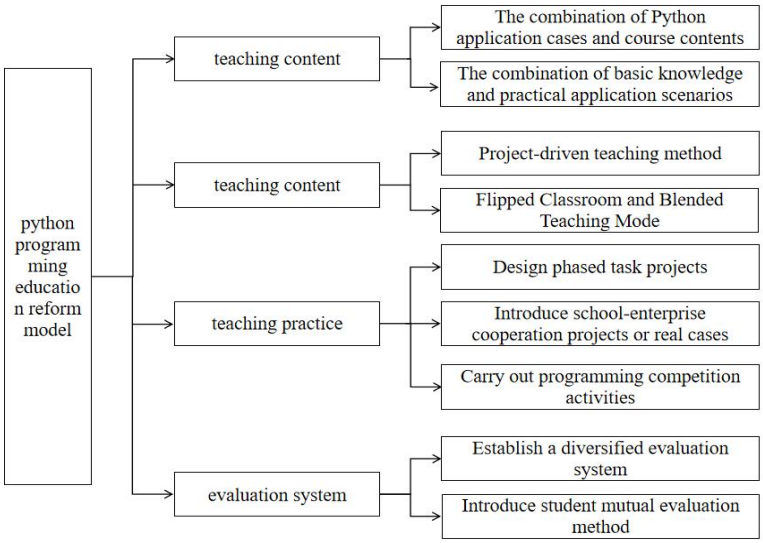

Figure 1. Content frame diagram

3.2 Combine basic knowledge with practical application scenarios. In order to improve students' practical programming ability, the course content should be changed from simple basic knowledge teaching to practical application scenarios. In the teaching process, we should not only pay attention to the basic knowledge of Python syntax, data structures and algorithms, but also let students master how to apply theoretical knowledge to practice through practical problem solving.

First, courses can set up project-driven learning models that combine the basics with practical problems. For example, after learning basic grammar, students can consolidate their knowledge and deepen their understanding by designing a simple automation tool or developing a small Web application. Through these projects, students can not only translate programming knowledge into the ability to solve practical problems, but also experience the practicality and fun of programming.

Secondly, the course can introduce interdisciplinary application scenarios, so that students can see the diversity and universality of Python in practical applications. For example, when learning data structures and algorithms, Python can be combined with practical applications in finance, medical care, education and other fields to explore how to solve practical problems in these industries with programming [15]. Through these application scenarios, students can deeply understand the advantages of Python and stimulate their interest in learning.

3.3 Project-driven teaching method. Project-Based Learning (PBL) is a teaching method with practical projects as the core, emphasizing that students can solve practical problems by participating in the development of real projects, so as to improve their programming ability, problem solving ability and teamwork ability [16]. In the Python programming course, PBL can enable students to understand the application of programming language in practice, and can deeply grasp the practical application of Python in data analysis, artificial intelligence, Web development and other fields.

In practical implementation, the course can design multiple phased project tasks, such as starting with a simple programming task and gradually transitioning to designing a small database application, developing a data analysis tool, or building a machine learning model. Through project-driven, students can not only learn the grammar and algorithm of programming language, but also experience the whole process of software development, such as requirement analysis, code implementation, testing and debugging, and enhance practical ability. In addition, PBL can stimulate students' interest in active learning, help them transform theoretical knowledge into practical abilities, and enhance their ability to solve complex problems.

3.4 Flipped classroom and blended teaching mode. The combination of flipped classroom and blended teaching model provides students with more opportunities for independent learning and interactive communication. In this model, students learn basic concepts and fundamentals on their own through online learning platforms, instructional videos and reading materials before class, while class time focuses on discussions, problem solving, collaborative projects and hands-on activities.

Specifically, the course can use flipped classroom to let students preview Python syntax, algorithms, and basic applications before class, and provide after-class exercises and instant feedback through an online platform to ensure that students can master the basics. In class, teachers can organize students to have group discussions, solve problems encountered by students in the learning process, carry out programming practice, or let students deepen their understanding of Python in the real development process through project cooperation. In this way, students can not only master knowledge points more flexibly, but also enhance their self-study ability and teamwork spirit.

3.5 Problem-oriented instructional design. Problem-Based Learning (PBL) emphasizes promoting learning by guiding students to solve practical problems. In Python programming courses, students can learn programming knowledge while solving problems by designing real-world problem situations, and learn how to apply programming skills to real problems.

For example, teachers can set challenging programming problems or practical projects, and students can gradually master Python programming techniques by analyzing the problems, developing solutions, implementing code, and finally completing the project. PBL not only helps students deepen their understanding of Python syntax and tools in problem solving, but also exercises their critical thinking and creativity. In addition, PBL can promote students to think actively, cultivate their ability to solve problems independently, and lay a solid foundation for future work and scientific research.

Through the innovation of these three teaching methods, Python programming courses will pay more attention to the cultivation of practicality, interactivity and problem solving ability. In this teaching mode, students will not only master programming skills, but also enhance their innovative thinking and comprehensive abilities through the experience of collaboration, discussion and practical projects.

3.6 Increase the proportion of practical courses. In Python programming courses, the combination of theoretical learning and practical operation is crucial. In order to enhance students' practical ability, the curriculum should increase the proportion of practical courses, and design task projects in stages, so that students can improve their programming level in the process of gradual deepening. Tasks at each stage should have clear learning objectives and practical applications, covering all aspects of Python programming, from basic syntax applications to complex data analysis, machine learning, Web development and other practical projects.

For example, the early stages can help students consolidate their basic knowledge of Python through simple programming tasks (such as calculators, sorting algorithms, etc.); In the middle stage, students can design some more challenging projects (such as student management systems, data processing tools, etc.), so that students can start to contact with practical applications; Advanced stage can introduce complex projects, such as data analysis and visualization, machine learning model building, etc., to cultivate students' comprehensive ability. Each project should encourage students to think independently and solve problems, while encouraging students to code optimization and function expansion to further enhance their programming skills and innovative awareness.

3.7 Introducing school-enterprise cooperation projects. In order for students to better understand the needs of the industry and the application of technology, the course can introduce school-enterprise cooperation projects or real scenario cases. By working with companies to provide students with the opportunity to actually participate in projects, students can be exposed to the programming techniques and project needs actually used by enterprises, and develop the ability

to solve real problems. Companies can provide students with data sets, business requirements, and even guide students through project development to enhance students' hands-on experience.

For example, schools can work with technology companies to invite enterprise engineers to participate in teaching and provide common Python application scenarios in the industry, such as big data analysis, machine learning, automated script development, and so on. Students can develop projects based on these real questions, resulting in a better understanding of the links between theory and practice. In addition, enterprises can provide some open source projects or actual needs, and students can directly connect with the needs of enterprises when participating in them, and improve their employment competitiveness.

3.8 Carry out programming competitions. Programming competition is an effective way to stimulate students' interest in programming and innovative ability. By holding programming competitions (such as ACM and LeetCode competition), students can demonstrate their programming ability in fierce competitions and improve their ability of teamwork, problem solving and innovative thinking. Such activities can not only enhance students' practical programming experience, but also promote their interaction and communication, and enhance the spirit of collective cooperation.

The school can hold internal programming competitions regularly to encourage students to challenge programming problems of different difficulty and solve practical problems; At the same time, the school offers a wealth of project development opportunities, allowing students to work as a team within a limited time to complete a small project or solve a specific problem. Through these activities, students are not only able to exercise their programming skills, but also develop the ability to solve problems quickly under pressure, thereby enhancing their interest in programming and gaining confidence in their future careers.

3.9 Establish a diversified evaluation system. In order to comprehensively evaluate students' learning results in Python programming courses, a diversified evaluation system should be established, combining process evaluation and outcome evaluation. This comprehensive assessment focuses not only on the final outcome of the student, but also on the ability and progress demonstrated by the student during the learning process.

The process assessment focuses on the development of students' skills in the learning process, including class participation, assignment submission, laboratory operation, project progress, and group cooperation. Through regular feedback and guidance, teachers can find the weak points of students in the learning process in time and help them make targeted improvement. In addition, the process evaluation can also check the students' mastery of knowledge points through regular quizzes, homework and experiment reports, and encourage students to constantly improve themselves.

Results-based evaluation is mainly aimed at students' final completed projects or exam results. Students' mastery of Python language and application skills is assessed by scoring their final works, programming projects, final exams, etc. Through the review of student projects, teachers can gain insight into students' innovative thinking, code quality, problem-solving skills, etc., ensuring that students not only master basic programming skills, but also have the flexibility to apply these knowledge in real-world problems.

3.10 Introduction of student mutual evaluation. In order to enhance students' sense of active participation and teamwork spirit, evaluation methods such as student mutual evaluation and project results display can be introduced, which can promote students' self-reflection and improvement and stimulate students' learning interest.

Student evaluations can effectively improve students' critical thinking about their own and others' projects. During project development, students can check on each other's code quality, suggest improvements, and evaluate the work of other teams. This mutual evaluation not only motivates students to pay attention to detail and improve the quality of their projects, but also helps them understand the importance of teamwork and communication. In addition, by evaluating others' projects, students can learn different solutions from them and stimulate more creative thinking.

The presentation of project results is another effective way of evaluation, which can promote the presentation of students' own work and the building of self-confidence. At the end of the course, students can show their achievements in the project through reporting, demonstration and other forms, including code implementation, function display and project summary. Through this demonstration, students can get feedback from teachers and classmates, and at the same time further enhance their understanding of the project and self-confidence. The presentation session can also encourage students to demonstrate their innovative ideas and technology applications, further stimulating their interest and motivation.

4. Summary

4.1 Research Conclusion. Through the research and practice of the teaching reform of Python programming course, remarkable results have been achieved and valuable experience has been accumulated. First of all, the updating of course content and the innovation of teaching methods have effectively improved students' learning interest and programming ability. The introduction of project-driven teaching, flipped classroom and problem-oriented learning methods enables students to apply what they have learned in real scenarios, enhancing their practical ability and innovative thinking. Through the operation of practical projects, students not only learn programming skills, but also improve the ability of teamwork and problem solving.

Secondly, curriculum reform promotes the overall improvement of students' comprehensive quality. The reformed curriculum pays more attention to the combination of basic knowledge and practical application, and in the process of solving practical problems, students not only improve their programming ability, but also cultivate innovative thinking and independent problem-solving ability. The development of programming competitions, hackathons and other activities has stimulated students' enthusiasm for learning and enhanced their sense of competition and achievement.

In addition, the reform of teaching resources and teacher training has also achieved good results. Through the development of open learning platform and the integration of resources inside and outside the school, students can obtain rich learning materials at any time, and further improve the learning effect. Through regular training and exchange of teaching experience, teachers not only improve their teaching level, but also better adapt to the new teaching mode and the application of technical tools.

4.2 Future Outlook. Although the teaching reform of Python programming course has achieved remarkable results, there is still room for further optimization and development. In the future, we will continue to deepen the reform of teaching content and methods, and explore more innovative teaching models and application scenarios to adapt to changing educational needs and technological developments.

First of all, continuous optimization of teaching content and methods is the core direction of future reform. As Python technology continues to evolve, emerging areas such as data analytics, machine learning, artificial intelligence, and more will continue to be an important part of the course content. We plan to bring these cutting-edge technologies into the classroom to help students better master the latest programming skills and applied knowledge. At the same time, teaching methods should be constantly innovative. We will pay more attention to problem-based learning (PBL) and project-driven teaching model to stimulate students' learning interest, enhance their practical application ability and innovation ability.

Secondly, exploring more innovative teaching models and application scenarios will be an important topic of future reform. On the basis of existing models such as flipped classroom and blended teaching, we will try to integrate virtual reality (VR), augmented reality (AR) and other technologies into classroom teaching to create a more immersive and interactive learning experience for students. In addition, with the continuous development of artificial intelligence and big data technologies, the use of these technologies for personalized teaching, learning analysis and intelligent assessment will also become an effective way to improve the quality of education. Through these innovative models, it is possible to better adapt to the learning needs of different

students and help them grow in a more diverse and open environment.

In short, the future development of Python programming courses will continue to keep up with technological progress and educational innovation, promote the continuous improvement of teaching content, methods and evaluation systems, provide students with more challenging and creative learning opportunities, and lay a solid foundation for their future career development.

References

[1] Ou Yu, Tian Xiaomei, Yu Xiaozhong. New exploration engineering teaching reform under the background of Python programming practice [J]. Computer knowledge and technology, 2024, 20 (8) : 156-158 + 180.

[2] Yang Keyang. Exploration on Reform of Teaching Mode of Python Programming Course [J]. China Management Information Technology, 2019,27(23):182-185.

[3] Zhao Yechun, Wang Senhong, Li Wanyi, et al. Research on Teaching reform of Python programming course enabled by Artificial Intelligence [J]. Information and Computer (Theoretical Edition),2024,36(21):1-6.

[4] Hu Yan-hua, Cui Yan-nan. Research and Practice on Teaching reform of Python Programming course under the background of new engineering [J]. Information and Computer (Theoretical Edition), 2019,36(21):224-226.

[5] Wang Xiaoning. Research on Teaching Paradigm Reform and Practice of Python Programming Course in Higher Vocational Colleges based on OBE concept [J]. Computer knowledge and technology, 2024, 20 (24) : 169-171 + 174.

[6] Sun Tongfeng, Wang Guanjun, Du Wenliang. Exploration of Practical Course Teaching based on Ideological and political Drive -- Reform of Python programming practice teaching [J]. Laboratory research and exploration, the lancet, 2024 (7) : 187-191 + 233.

[7] Chen Hao-qiang, Liang Zhen, Liu Fang, et al. New engineering background of Python programming teaching reform [J]. Computer knowledge and technology, 2024, 20 (17) : 141-144.

[8] Su Tong. Analysis on Teaching reform of Python Programming Course for Artificial Intelligence [J]. Office Automation, 2019,29(09):47-49.

[9] Gong Ping, Liu Hui, Liao Yifan. Python programming major basic curriculum expansion teaching reform [J]. Journal of Hunan Polytechnic of Posts and Telecommunications,2023,22(04):89-93. (in Chinese)

[10] Luo Mengzhen. Course education under the background of Python programming technology to explore teaching reform practice [J]. Computer knowledge and technology, 2023 (25) : 146-149.

[11] Zhang Lei, Li Hongan, Li Xia, et al. Teaching reform of "One Center, Two Interaction and six Ways" in Python programming under the background of new engineering [J]. Computer education, 2023, (8) : 86-90.

[12] Zhang Minglu, Wang Lili. Reform and practice of "Online + Offline" mixed teaching of Python Programming Course [J]. Information Systems Engineering,2023,(07):153-156.]

[13] Yue Hongwei, Fan Yani, Zhang Hui, et al. Python application of artificial intelligence course teaching reform and practice of analysis [J]. Computer knowledge and technology, 2023, 12 (16) : 168-170.

[14] Liu Jingle, Chang Xiaohong, Jian Weiping, et al. Military medical university in Python reform and exploration of the teaching mode [J]. Computer knowledge and technology, 2023 (15) : 135-138.

[15] Zhou Xu. Exploration on Teaching reform of "Fundamentals of Python Programming" under Blended teaching mode [J]. Education and Teaching Forum,2023,(18):49-52.

[16] Liu Zhaohong. Python language teaching reform under the concept of OBE-TOPCARES [J]. Fujian computer, 2023, 33 (02) 6:121-124.