Exploring Innovations in Teaching Reform for Python Programming Under Engineering Education Accreditation

Yumei Lin and Lianhua Fang
Quanzhou University of Information Engineering, Quanzhou, China

Keywords: Engineering Education, Innovation, Teaching Reform, Python.

Abstract: Engineering education serves as a crucial pathway for training engineering talent and driving technological innovation and social progress. However, traditional teaching models in engineering education suffer from various issues, such as a detachment of theory from practice, outdated teaching content, and limited teaching methods. These problems fail to meet the demands of modern industry and societal development. Therefore, this study aims to explore innovative approaches to empower teaching reforms in engineering education. Using the example of "Python Programming" instruction, we investigate how innovative ideas can be applied to enhance the teaching of Python programming, thereby improving students' practical skills and fostering their innovative thinking abilities. The findings of this research will provide valuable insights for cultivating highly qualified engineering professionals.

1 INTRODUCTION

With the emergence of new technologies and the continuous evolution of society's demand for engineering talents, engineering education reform needs to maintain close contact with social demand. Deepening the reform of engineering education and teaching can promote the improvement of education quality. Cultivating students' innovation, practical and lifelong learning abilities has become a key point of education reform. China's engineering education reform has made significant progress in promoting engineering education reform, including efforts in curriculum reform, teaching method innovation, and increasing the evaluation of practical aspects. Zhu Weiwen (Zhu Weiwen, 2023) studied the new trends in international engineering education reform from the changes in GAPC. The four new trends in international engineering education reform include strengthening the concept of sustainable development, interdisciplinary integration of engineering ethics and cultural diversity, focusing on learning outcomes, and being associated with the registration and certification system. Qian Fangfang (Qian Fangfang, 2021) proposed six measures for engineering education reform from the perspective of the challenges faced by local undergraduate engineering education reform under the background of "New Engineering", including improving talent cultivation, enhancing teacher's quality, promoting curriculum reform, etc. Zhu Qing (Zhu Qing, 2022) analyzed and interpreted the connotation and characteristics of complex engineering problems by comparing the certification standards for engineering education in China, America, France, and Germany. Engineering education reform is promoted through four aspects including goal cultivation, curriculum reform, teaching innovation and assessment evaluation. More and more universities are paying attention to interdisciplinary cooperation to provide more comprehensive engineering education by integrating knowledge and skills from different fields. Xu Lihui (Xu Lihui, 2020) explored the talent cultivation model of interdisciplinary engineering education by conducting exploratory case studies on engineering education reforms carried out by University College London, Aalborg University, Princeton University, Massachusetts Institute of Technology, University of Toronto, and McMaster University in recent years. Based on the background of "New Engineering", Wan Chao (Wan Chao, 2022) discussed the construction of interdisciplinary courses in mechanics. Li Jinhuan (Li Jinhuan, 2022) studied the path of interdisciplinary course construction in engineering colleges. Python has been widely used in engineering education to help students better understand and apply relevant knowledge.

Computer programming is an important foundation course in the field of software engineering,

Lin, Y. and Fang, L.
Exploring Innovations in Teaching Reform for Python Programming Under Engineering Education Accreditation.
DOI: 10.5220/0012283900003807

In *Proceedings of the 2nd International Seminar on Artificial Intelligence, Networking and Information Technology (ANIT 2023)*, pages 355-361
ISBN: 978-989-758-677-4

and it is also one of the core competencies required for software engineering professionals. Python, as a simple and powerful programming language, has been widely used in various fields. However, traditional Python programming teaching methods are often too theoretical, lack practical exercises and innovative thinking training, and are difficult to inspire students' interest and creativity. Therefore, innovative teaching methods for "Python Programming" have been developed to combine practice with theory, cultivate students' innovative thinking and practical abilities, and improve the quality and level of engineering education and teaching, which has important practical significance and development value. Some computer science educators have proposed course reforms based on engineering certification, such as data structures (Li Zhaokui, 2019), linear algebra (Wang Haibo, 2019), computer introduction (Zhang Xiaoming, 2019), computer English (Ma Changxia, 2020), etc. Some scholars have proposed Python course teaching reforms based on the OBE concept (Lu Keqing, Zhou Jian). Some scholars have also proposed research on course achievement assessment methods (Bai Yanhong - Ouyang Hongji).

2 INNOVATION EMPOWERS ENGINEERING EDUCATION

Engineering education is an applied education (Lin Jian, 2015). In order to meet the needs of society and the economy, it aims to cultivate students' practical skills, problem-solving abilities, innovative thinking, and team spirit. Engineering education originated in Europe. In the early 20th century, engineering education began to expand into civil fields. With the rapid development of science and technology and the rapid development of the social economy, engineering education gradually shifted from traditional theoretical teaching methods to practical teaching methods, focusing on cultivating students' practical application abilities and innovative thinking. Modern engineering education emphasizes the cultivation of students' comprehensive qualities, emphasizes practical operation and innovative ability training, and integrates interdisciplinary cooperation and international perspectives into teaching.

Innovative teaching empowers engineering education, with a practice-oriented approach that emphasizes practical teaching and combines theoretical knowledge with practical applications to cultivate students' practical skills and problem-solving abilities. Students are encouraged to undergo innovative thinking training to develop their creativity and innovation skills, and to improve their overall qualities. Students are also trained in teamwork and communication skills. Different teaching methods and approaches are adopted based on students' personalities and characteristics to improve teaching effectiveness.

3 CURRENT SITUATION AND PROBLEMS OF PYTHON PROGRAMMING TEACHING

The teaching of Python programming has gradually gained attention in recent years, as Python language has consistently ranked first in the TIOBE index. However, there are still some problems in Python programming teaching, such as outdated teaching materials that lack ideological elements, making it difficult to achieve ideological education goals. The teaching method is too teacher-centered, with knowledge-based lectures that fail to attract students' attention and stimulate their creativity and innovative development. The lack of diversity and personalization in the classroom makes it difficult to inspire students' interest and innovation abilities. The practical projects are too standardized, failing to attract students' attention or stimulate innovative development, and lacking engineering applications, making it difficult for students to master practical application skills and meet the needs of talent cultivation for applied professionals. The evaluation system is too simplistic, with only regular and final exam scores used as evaluation standards. This fails to accurately measure students' learning outcomes, and the evaluation data cannot be effectively digitized or humanized for management purposes, making it difficult to conduct teaching reflection and continuous improvement.

In summary, there are still some problems in Python programming teaching, despite its importance in fields like artificial intelligence, big data, and cloud computing. Python programming teaching needs to adapt to industry demands and cultivate talent that meets market needs. Innovative teaching methods and approaches are needed to improve the quality and level of teaching, emphasizing practical teaching and engineering applications to cultivate students' practical skills and problem-solving abilities. Students' learning interests and innovative abilities should be stimulated to improve learning outcomes and achievements.

4 INNOVATIVE IDEAS AND MEASURES

Based on students' individual differences and advantages, we will focus on cultivating socially recognized and applied talents that meet society's needs. Adhering to the educational philosophy of "morality first and OBE," we will explore curriculum reform based on students' learning situations and teaching pain points, promoting innovation in teaching from teaching philosophy, teaching content, teaching methods, and teaching evaluation.

4.1 Reform Teaching Philosophy

Adhering to the educational philosophy of "morality first, student-centered, problem-oriented, inquiry-based learning, continuous improvement," this course will build a learning community for teachers and students, emphasizing teacher-student interaction, mutual learning, collaborative innovation, and win-win cooperation. We will adhere to the "student-centered" and "outcome-based education" (OBE) concepts to continuously improve the quality of teaching. We will permeate the "Internet +" mindset, fully utilize rich teaching resources such as spoc, smart classrooms, learning platforms such as PTA for programming experiments, creating a self-directed and collaborative learning environment, and achieving hybrid teaching models that integrate online and offline, in-class and out-of-class. We will focus on students' needs, pay attention to classroom feedback, and adjust teaching strategies in a timely manner, emphasizing students' knowledge acquisition outcomes. We will focus on cultivating students' self-learning ability and flexible application of knowledge ability while integrating ideological education elements (such as the power of role models, the spirit of the times, moral cultivation, and socialist core values), silently cultivating students' scientific spirit, innovation consciousness, patriotism, and social responsibility. Based on spoc learning and the effective teaching structure of BOPPPS, we will focus on improving teaching effectiveness, creating an efficient Python classroom, fostering a harmonious teaching atmosphere, guiding teachers to teach well and students to enjoy learning, and guiding students to actively "think, discuss, question, practice, explore, understand reasoning and apply knowledge," achieving the three-in-one teaching goals of value shaping, ability cultivation, and knowledge exploration, and ultimately implementing the fundamental task of moral education.

4.2 Optimizing and Updating Teaching Content, Optimizing Teaching Design

1) Incorporating the "Internet+" mindset, optimizing and updating teaching content.

Python has extensive support in fields such as artificial intelligence, data analysis, automated operations and maintenance, cloud computing, and web crawling. Therefore, it has different applications in industries such as industry, agriculture, commerce, and healthcare. The "Internet +" is deeply integrated with all industries. By permeating the "Internet +" mindset and innovating teaching content, we can turn original knowledge points into exploratory learning tasks. The design of teaching content follows the principles of content modularization, case-based interestization, and ability layering. We will let students learn with questions and think with questions. Python is constantly updated and upgraded with many new features added in new versions. We will explore these new features with students and incorporate them into teaching content. The model of teaching is shown in the figure 1 below.

2) Reshaping teaching models and optimizing teaching design.

The exploration of a "hybrid online and offline + BOPPPS" teaching model that meets the needs of talent cultivation and students' actual levels, providing a strong platform support and operational drive for integrating teaching content, teaching methods, and educational goals. This new teaching format ensures a concentrated time for discussing case integration in the existing class hours, and increases practical operation links.

Students can independently understand the knowledge points online, and adopt advanced teaching methods such as inquiry-based learning in offline face-to-face classes, which can be achieved in small-class learning. Students will discuss and summarize the course content, draw mind maps, and complete high-level teaching activities such as group projects based on the content of this course. Through high-level teaching activities, students' strong curiosity and thirst for knowledge are stimulated, guiding them to achieve the transformation from "I have to learn" to "I want to learn" to "I can learn" to "I can learn well." We will work with other colleges to create spoc courses with ingenuity, providing a resource carrier for deep integration of online and offline hybrid teaching. The online spoc teaching resources are rich, mainly including six core contents: teaching videos, teaching courseware, in-class quizzes, typical exercises, chapter quizzes, etc.

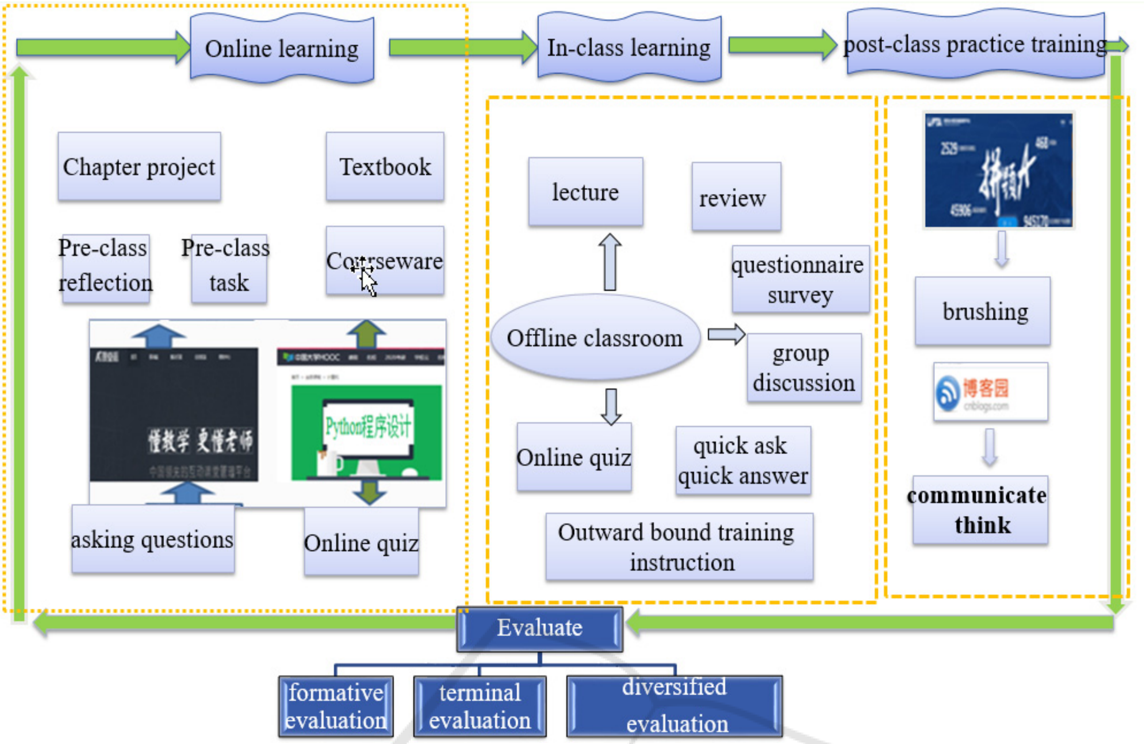

Figure 1. Model of teaching.

3) Adopting a problem-oriented approach and using inquiry-based teaching.

Promoting teaching by competition, promoting learning by competition, teaching and learning. Guiding students to carry out group cooperation inquiry learning after class, and take the initiative to think, question, discuss, consult literature, penetrate professional thoughts, reflect the frontier of the discipline. Guiding students to carry out python project analysis, programming and testing, prepare PPT for defense, and carry out project discussion. Making use of software engineering professional associations to cultivate students' independent learning, research and innovation ability, teamwork spirit and communication and expression ability, so that students can stimulate their interest in personal experience, understand reason, and apply what they have learned. I really feel that learning programming is "interesting, relevant and useful".

Linking theory with practice and make programming closer to life. For example, when teaching "branch structure," we will introduce the calculation of electricity bills and the problem of grading 100 points into 5 levels. The examples used in class are familiar examples around students and classic cases. By analyzing the code written by senior students, we guide students' thinking about branching and increase their understanding of knowledge.

Paying attention to the ideological education of the course and give equal weight to moral education and intellectual education. To teach others, one must first teach oneself. Teachers are guides and companions for students' learning. The rigorous scientific literacy of teachers, their dedication to teaching and educating people, and their pursuit of teaching art are the best elements of ideological education in the course. In addition to focusing on subject knowledge and skills, processes and teaching methods, this course will fully tap into the collective wisdom of the teaching team, forming a joint force for education. We will design ideological education elements from small-class discussions on ideological elements to sharing at the college level, and finally concentrate on group discussions to silently infiltrate ideological education elements into the entire process of Python programming teaching and learning.

4.3 Making Full Use of Professional Associations and Software Subject Competition Bases

Using competitions to promote teaching and learning, and guide students to engage in group cooperative inquiry learning outside of class, actively think, question, discuss, and consult literature to penetrate professional ideas and reflect the forefront of the discipline. We will guide students to analyze Python projects, program and test, make defense PPTs, conduct project discussions, and use software engineering professional associations to cultivate students' independent learning, research and innovation ability, teamwork spirit, and communication skills. We will inspire students' interests and truly feel that learning programming is "interesting, feasible, and useful."

4.4 Using PTA for Problem-Solving Practice to Improve Students' Programming Skills

Using high-quality exercise resources from various schools, select and update them. After-class exercises and quizzes will be conducted online using PTA for smart analysis, and students' homework completion will be automatically judged to strengthen the learning of memorization and comprehension-based knowledge, enhance low-level cognition, and timely discover students' weak points in learning to modify teaching strategies and improve teaching quality.

Teachers and students jointly build cases to achieve teaching coupling iteration (Li Zhiyi, 2014). Using seminar-style teaching methods, teachers act as "moderators" and guide student teams to collaborate on case construction based on demonstrated case construction. Both teaching and learning stages involve case design and optimization, prompting students to use knowledge to actively complete more complex integration processes, guide students to think about and explain ideological content in the course, and strengthen mid-level cognition.

4.5 Strengthen Process-Based Assessment and Design a Diversified Evaluation System

Strengthening process-based assessment based on learning data analysis, dynamically improve teaching strategies, and provide timely feedback to students. The course grade is composed of three parts: ① online MOOC learning (10%): including in-class quizzes, discussion forum reply frequency, and questions asked; ② offline classroom performance (20%): including flipped classroom interactive answering, mind map summary, experiments, and innovation rewards; ③ PTA online testing (20%): including periodic testing, computer lab test scores, etc.; ④ offline final exam (50%). The diversified teaching evaluation system is shown in the table below.

Table 1. Assessment indicators.

Online MOOC learning (8%)		Offline classroom performance (22%)			PTA online testing (20%)		Offline finalexam (50%)
In-class quizzes	Discussion forum reply frequency and questions asked	Interactive answering	Mind map summaries	Experiments and innovation rewards	Periodic tests	Computer lab tests	Final exam
5%	3%	5%	5%	12%	10%	10%	50%

4.6 Build a Hierarchical and Normalized Teacher Teaching Ability Training System

To improve teachers' teaching and educating abilities and build a first-class teaching team, we will implement a system of young teacher mentors, old teacher mentoring, and normalized collective teaching research activities. We will carry out collective lesson preparation, teaching observation, lesson presentation, and special topic discussions to pool our resources, share knowledge, and improve together, cultivating a team spirit and emphasizing collaborative development. We will actively participate in teaching competitions to promote teaching, improve, exchange, and assist excellence. We will apply for teaching reform projects, conduct academic research on teaching, strengthen curriculum construction, innovate teaching models, reform teaching methods, integrate teaching with research, and promote teaching with research. We will host computer teaching forums, blockchain conferences, etc., implementing the concept of "inviting in, going out, and more communication," expanding our horizons and learning advanced educational teaching concepts. We will gradually guide young teachers from teaching novices to backbone teachers, and ultimately to expert teachers.

Through these measures, the quality of this course has been significantly improved, and students' learning enthusiasm and achievements have also been significantly improved.

5 ANALYSIS OF COURSE OBJECTIVE ACHIEVEMENT AND CONTINUOUS IMPROVEMENT

Course objective 1 is assessed through in-class quizzes, discussion forum reply frequency and questions asked, mind map summaries, periodic test scores, offline final exam scores, and some multiple-choice and fill-in-the-blank questions in the final exam. It mainly tests the mastery of data types and program structures. Course objective 2 is assessed through in-class quizzes, interactive answering, experiments and innovation rewards, periodic test scores, computer lab test scores, offline final exam scores, daily homework, practical reports, and some programming questions in the final exam. It mainly tests the mastery of various Python knowledge points. Course objective 3 is assessed through discussion forum reply frequency and questions asked, interactive answering, experiments and innovation rewards, computer lab test scores, and some programming questions in the final exam. It mainly tests the integration and application of Python, as well as students' abilities in self-learning and innovation. The course objectives, assessment methods, and corresponding proportions are shown in Table 2.

Table 2. Course objectives, assessment methods, and corresponding proportions.

Course Objective	Assessment Method and Weight								Weight
	In-class quizzes	Discussions	Interactive answering	Mind map summaries	Experiments &innovation rewards	Periodic tests	Computer lab tests	Final exam	
target 1	2	2	0	5	0	6	0	15	30
target 2	3	0	2	0	6	4	4	15	34
target3	0	1	3	0	6	0	6	20	36

Based on Table 2, calculate the achievement of each course objective. Each assessment method is weighted to a total score of 100. Taking course objective 1 as an example, the achievement is calculated as follows: Achievement = (0.02 × in-class quizzes + 0.05 × discussion forum reply frequency and questions asked + 0.02 × mind map summaries + 0.06 × periodic tests + 0.15 × final exam score). The final achievement of the course objectives is the minimum of the achievement of each course objective i. The achievement of course objective i is calculated as$\frac{\sum_1^3 x_{ij}w_{ij}}{\sum_1^3 y_{ij}w_{ij}}$ where xij is the average score of the jth assessment method for objective i, yij is the expected score of the jth assessment method for objective i, and wij is the weight of the jth assessment method for objective i, where j=1,2,3.

Taking the scores of 62 students from Class 1 of the 2020 Software Engineering Major as a sample, the achievement of the course objectives is as follows: the achievement of objective 1 is 0.86, the achievement of objective 2 is 0.82, and the achievement of objective 3 is 0.73. All are higher than 0.7, indicating that students have mastered the main teaching content of the course and achieved the training objectives of the course.

The achievement of course objective 3 is the lowest, and through analysis, three problems were identified: (1) Some students are not accustomed to using programming thinking to analyze and solve problems. In terms of "combination and construction" (analyzing cases and programming using professional basic knowledge), their ability has not been effectively trained and improved. (2) Some students have not truly understood and integrated knowledge points. The next step in improvement measures is: (1) Continuously pay attention to the development trends of Python, increase knowledge content corresponding to new technologies, and complete corresponding teaching resource construction. (2) Timely forward excellent technical articles from relevant public accounts to students. (3) Explore more realistic and effective student evaluation plans. (4) Optimize experimental content, appropriately reduce verification experiments, strengthen comprehensive, design-oriented, and innovative experimental teaching. Expand comprehensive and innovative practical content based on new research projects and engineering applications.

6 CONCLUSION

In this paper, we focus on the problem of reform and innovation of Python programming teaching under the background of engineering education certification. Firstly, it introduces the innovative empowerment of engineering education certification, discusses the current situation and problems of Python programming courses, and focuses on the innovative ideas and reform measures in course construction, transforming Python programming classrooms from traditional classrooms to smart classrooms, knowledge-based classrooms to ability-based classrooms, indoctrination classrooms to practical classrooms, and closed classrooms to open classrooms. By using smart teaching tools and modern information methods, students who are not active or lack a sense of participation are guided to stimulate their enthusiasm for autonomous learning. Teaching reform and innovation need to be further refined and continued to promote students' comprehensive development with the concept of "awakening, empowerment, growth, talent, and maturity".

ACKNOWLEDGMENTS

This work was financially supported by funds (1) Fujian Province Undergraduate Education and Teaching Reform Research Project (FBJG20220181) (2) Research Project on Vocational Education and Teaching Reform of Fujian Vocational and Technical Education Center (GB2020036) (3) Fujian Provincial Education Department of Young and Middle-aged Teachers Education Research Project (JAT210556) and so on.

REFERENCES

Zhu Weiwen, Chen Yiyi, Li Yadong. New trends in international engineering education reform seen from changes in GAPC (J). *Higher Education Development and Evaluation*, 2023, 39(02): 40-49+120-121.

Qian Fangfang, Zhi Fengsheng. Analysis of reform strategies for engineering education in local undergraduate colleges (J). *University*, 2021, No. 541(47): 88-90.

Zhu Qing. Analysis of the reform path of engineering education-based on the comparative study of engineering education certification standards in China, the United States, France, and Germany (J). *Modern Commerce and Industry*, 2022, 43(21): 31-33. DOI: 10.19311/j.cnki.1672-3198.2022.21.013.

Xu Lihui, Wang Sunyu. A new model of interdisciplinary cooperation for engineering talent cultivation-exploratory multi-case study of engineering education (J). *Tsinghua University Journal of Education*, 2020, 41(05): 107-117. DOI: 10.14138/j.1001-4519.2020.05.010711.

Wan Chao, Yu Yang, Huo Bo. Construction and discussion of interdisciplinary courses in mechanics under the background of new engineering (J). *Mechanics and Practice*, 2022, 44(02): 379-384.

Li Jinhuan. Research on the path of interdisciplinary course construction in engineering colleges and universities under the background of the new engineering (J). *Harbin University of Science and Technology*, 2022. DOI: 10.27063/d.cnki.ghlgu.2022.000172.

Li Zhaokui, Wang Yan, Wan Runze, et al. Research and practice on formative evaluation of "Data Structure Course Design" for engineering education certification(J). *Software Guide (Education Technology)*, 2019, 18(10):64-65.

Wang Haibo, Wen Haiying, Guo Meizhen. Construction and exploration of linear algebra course under the background of computer engineering education certification (J). *Computer Knowledge and Technology*, 2019, 15(6):89-90.

Zhang Xiaoming, Wei Zhanhong. Teaching reform and practice of introduction to computer science based on engineering education certification (J). *Computer Education*, 2019, 17(11):99-103.

Ma Changxia. Teaching reform of computer professional English course under the background of engineering education certification (J). *University Education*, 2020, 9(3):111-114.

Lu Keqing, Wang Wen, Yang He. Formative evaluation method of course objectives based on OBE-taking "Engineering Drawing" as an example (J). *Education Forum*, 2020, 12(8):173.

Zhou Jian, Liu Jianli, Fu Jiajia, et al. Practical exploration of course attainment evaluation of textile engineering specialty under the background of engineering education certification(J). *Textile and Garment Education*, 2019,34(6):495-497.

Bai Yanhong. Research and practice on formative evaluation of course objectives under the background of engineering education certification (J). *China Higher Education Research*, 2019, 33(12):60-63.

Yu Bingjun, Guo Fen, Zhang Yongxiong, Wang Liangming. Teaching reform of Python language programming course based on OBE concept (J). *Theoretical Research and Practice of Innovation and Entrepreneurship*, 2023, 6(01):41-43+47.

Yang Feifei, Wang Juan, Zhang Yue. Syllabus for Python data analysis and visualization course in data science and big data technology major under the background of engineering education certification (J). *Computer Knowledge and Technology*, 2021, 17(03):33-36. DOI: 10.14004/j.cnki.ckt.2021.0112.

Xu Shengzhou, Du Xiaokun. Course objectives attainment evaluation method for "JavaEE Program Design" based on the concept of engineering education certification (J). *Modern Computer Science and Technology*, 2021(19):170-174.

(xi) Ouyang Hongji, Ge Meng, Song Xiaoxue. Teaching reform of JavaEE application development course under the background of engineering education certification (J). *Journal of Science Education at Universities*, 2023, 43(06):95-102.

Lin Jian. Engineering education certification and the reform and development of engineering education (J). *Research on Higher Engineering Education*, 2015(2):10-19.

Li Zhiyi, Zhu Hong, Liu Zhijun, Xia Yuanjing. Guiding the Reform of Higher Engineering Education Teaching with the Achievement-Oriented Education Concept (J). *Research in Higher Engineering Education*, 2014, (02):29-34+70.