Journal of Computer Science and Electrical Engineering
Print ISSN: 2663-1938
Online ISSN: 2663-1946
DOI: https://doi.org/10.61784/jcsee3075

TEACHING METHOD REFORM OF PYTHON PROGRAMMING COURSE EMPOWERED BY AI TECHNOLOGY

Yi Han
Jiayang College, Zhejiang Shuren University, Hangzhou 310009, Zhejiang, China.
Corresponding Email: 187267654@qq.com

Abstract: In the tide of the digital era, programming ability has become one of the essential core competencies for students in many majors. Python language, with its concise and readable syntax, powerful and rich libraries, and wide range of application fields such as Web development, data analysis, artificial intelligence, and machine learning, occupies an important position in the field of programming education. The traditional teaching mode of Python programming courses has gradually revealed certain limitations in improving students' interest in programming, learning effects, and cultivating the ability to solve practical problems. With the rapid development of artificial intelligence (AI) technology, its application in the field of education is constantly expanding and deepening.
Keywords: Python; Programming; Artificial Intelligence; Teaching method; Reform

1 INTRODUCTION

AI technology has the significant characteristics of intelligence, personalization, and strong interactivity, which can bring innovative changes to the teaching of Python programming courses. It can provide personalized learning paths and precise learning support according to students' learning situations and characteristics; reduce the difficulty of programming learning and enhance students' learning experience through intelligent tools; stimulate students' learning interest and initiative by using rich interactive means. Therefore, exploring the teaching reform of Python programming courses empowered by AI has important practical significance and urgency, which is expected to inject new vitality into programming education and improve the quality of teaching and talent training.

2 CURRENT SITUATION AND ANALYSIS OF TRADITIONAL PYTHON PROGRAMMING TEACHING METHODS

2.1 Teacher-Centered Teaching

Most traditional Python programming courses adopt the teaching mode where teachers explain theoretical knowledge and demonstrate code examples in class, and students practice after class. In this mode, teachers are in a dominant position, and students passively accept knowledge. For example, when explaining the definition and use of Python functions, teachers usually first introduce the syntax structure of functions, then demonstrate through several simple function examples, and students imitate and practice according to the teacher's steps. Throughout the process, students lack the opportunity to think and explore actively, and it is difficult to fully understand the connotation and application scenarios of knowledge, resulting in low learning enthusiasm. According to relevant surveys, about 60% of students think that traditional classroom teaching is boring and lacks a sense of participation [1].

2.2 Lack of Personalized Teaching

Due to differences in students' learning foundations, learning abilities, and learning styles, unified teaching content and progress are difficult to meet the needs of each student. In traditional teaching, it is often difficult for teachers to provide personalized guidance according to the characteristics of each student. For example, students with a good foundation and strong learning ability may find the teaching content too simple and the learning progress slow, leading them to be easily distracted in class. For students with a weak foundation, they may encounter difficulties in understanding complex programming concepts and code logic, fail to keep up with the teaching progress, and gradually lose confidence in learning [2].

2.3 Single Form of Resources

The learning resources of traditional Python programming courses mainly include textbooks, courseware, and a small number of online videos. Textbook content is usually systematic, but the form is relatively single, lacking vividness and interactivity; courseware is often a simple refinement of textbook content, which is difficult to attract students' attention; the number of online videos is limited, and the quality of some videos is uneven, which cannot fully meet students' learning needs. For example, when learning Python's data analysis library pandas, textbooks may only describe the functions and methods of the library in words, and it is difficult for students to understand its practical application through static words [3].

2.4 Lack of Dynamic Updates

With the continuous development of Python technology and the increasing enrichment of application scenarios, new libraries, frameworks, and programming methods are constantly emerging. However, the update speed of traditional learning resources is slow, which cannot timely reflect the latest developments in the industry and the trend of technological development. This makes the knowledge learned by students may be disconnected from practical applications, and it is difficult for them to quickly adapt to the needs of job positions after graduation. For example, in recent years, Python's application in the field of artificial intelligence has made significant breakthroughs, but some textbooks and teaching resources still have little introduction to relevant content [4].

2.5 Lack of Authenticity in Practical Projects

Projects in traditional practical teaching are often designed to practice a certain knowledge point or skill, which are quite different from actual engineering projects. When completing these projects, students find it difficult to experience the complexity and challenges in real project development, and cannot effectively cultivate the ability to solve practical problems. For example, when learning Python Web development, practical projects may only simply build a static web page to realize basic page layout and link jump, without involving common problems in actual projects such as database interaction, user authentication, and performance optimization[5,6].

2.6. Insufficient Practical Guidance

In the process of practical teaching, due to the large number of students, it is difficult for teachers to provide detailed guidance to each student. When students encounter problems, they often cannot get effective help in time, leading to the accumulation of problems and affecting learning effects. For example, in Python project practice, students may encounter various syntax errors, logical errors, or environment configuration problems. If they cannot be solved in time, students may get into trouble and reduce their learning enthusiasm.

2.7 Single Evaluation Method

The evaluation of traditional Python programming courses mainly focuses on the final exam results, while the proportion of usual grades is relatively small. The final exam usually adopts the form of a closed-book exam, focusing on examining students' memory and understanding of theoretical knowledge, but insufficiently examining students' programming practice ability, innovation ability, and ability to solve practical problems. For example, in the final exam, there are a large number of multiple-choice questions, fill-in-the-blank questions, and short-answer questions, while the score of programming practice questions is low and the types of questions are relatively rare [7,8].

2.8 Lack of Process Evaluation

Traditional evaluation methods often only focus on students' final learning results, ignoring their performance and progress in the learning process. This makes it impossible for teachers to timely understand students' learning situation and adjust teaching strategies; students also find it difficult to find their own problems in the learning process and make targeted improvements. For example, in the usual learning process, students may have difficulties in mastering a certain knowledge point, but due to the lack of process evaluation, neither teachers nor students can detect it in time, leading to the gradual accumulation of problems.

3 RELATED CONCEPTS FOR TEACHING METHOD INNOVATION

3.1 Personalized Learning Theory

Personalized learning theory emphasizes providing customized learning paths and support according to the characteristics and needs of each student to meet students' diverse learning needs and improve learning effects. AI technology can deeply understand students' learning styles, knowledge mastery, learning progress, and other information through the analysis of students' learning data, so as to tailor personalized learning plans for students. For example, AI can judge students' weak links according to the types and frequencies of errors in programming exercises, and recommend targeted learning materials and exercises for them; it can also dynamically adjust learning plans according to students' learning progress to ensure that each student can learn at a pace suitable for themselves. This personalized learning support can give full play to students' learning potential and improve the efficiency and quality of learning.

3.2 Intelligent Education System

Intelligent education system is a system that uses artificial intelligence, computer science and other technologies to simulate the teaching process of human teachers and provide students with intelligent learning support. It mainly includes domain knowledge model, student model, and teaching strategy model. The domain knowledge model stores the relevant knowledge and skills of the course; the student model describes the student's learning state and

characteristics through the collection and analysis of student learning data; the teaching strategy model provides students with personalized teaching strategies and guidance according to the student model and domain knowledge model. In Python programming courses, the AI-based intelligent education system can realize real-time monitoring and analysis of students' learning process, and provide accurate teaching content and guidance according to students' specific situations. For example, when students encounter errors in writing Python code, the intelligent education system can quickly locate the cause of the error and provide corresponding solutions and learning suggestions, just like an intelligent teacher guiding students to learn at all times.

4 SPECIFIC MEASURES FOR TEACHING METHOD INNOVATION

4.1 Intelligent Programming Assistant

Intelligent programming assistant is a powerful tool based on AI technology, which can provide all-round support for Python programming learners. It is very necessary to develop and integrate a Python learning platform with an intelligent programming assistant to realize the integration of AI technology and the development environment (IDE). The built-in AI dialog box supports natural language interaction, and learners can generate corresponding Python code by describing their needs in natural language. When a learner inputs "Create a function to calculate the average of two numbers", the intelligent programming assistant will quickly generate the following code:

```
def calculate_average(num1, num2):
return (num1 + num2)/2
```

This intuitive operation method greatly reduces the threshold for programming beginners, allowing them to quickly complete basic tasks without spending a lot of time memorizing complex syntax. In the actual programming process, the AI engine of the intelligent programming assistant can also detect syntax errors and logical vulnerabilities in the code in real time and give clear and accurate modification suggestions. If a learner forgets to add a colon after an “if” statement, the assistant will promptly prompt "Missing colon, please add it". It can also identify redundant code and put forward optimization schemes to help learners develop good programming habits and improve code quality. For example, for an inefficient code that uses a simple loop to traverse a list for summation, the intelligent programming assistant can suggest using Python's built-in sum function for optimization, making the code more concise and efficient.

4.2 Personalized Learning Path Planning Using AI

AI technology can accurately evaluate learners' basic level through testing their initial programming knowledge, such as online programming challenges or questionnaires that include basic operations of programming languages, application of data structures, etc., which learners complete after registration on some programming learning platforms. According to the evaluation results, a personalized learning path is tailored for each learner.

For learners with zero foundation, AI may recommend starting with Python basic syntax, such as variable definition, data types, control flow statements, etc., with simple and easy-to-understand examples and exercises to help them gradually establish programming thinking. For example, first learn how to define integer, floating-point, and string variables, then write simple programs to realize variable assignment, operation, and output. As learning progresses, more advanced concepts such as functions and modules are gradually introduced. For learners with a certain foundation, AI will suggest that they study advanced topics in depth, such as algorithm design, object-oriented programming, or programming frameworks in specific fields. For example, for learners who have mastered Python basic syntax, they can be recommended to learn data structure and algorithm courses, and improve their programming ability by implementing various classic algorithms such as sorting algorithms and searching algorithms; or guide them to learn Python application frameworks in fields such as Web development, data analysis, and artificial intelligence, such as Flask, Django, pandas, TensorFlow, etc., to broaden their technical horizons and meet the development needs of different learners.

4.3 Development of AI-Based Automatic Homework Correction System

In order to improve the efficiency and accuracy of homework correction and reduce the workload of teachers, an AI-based automatic homework correction system can be developed. The system can automatically correct and feedback students' homework, and when necessary, use natural language processing and machine learning technologies to automatically evaluate students' submitted text homework and provide detailed feedback.

Teachers can set custom scoring standards according to course requirements and homework goals to adapt to different disciplines and homework types. When correcting Python programming homework, the scoring weights can be set in terms of the accuracy of code function implementation, code standardization (such as naming rules, indentation format, etc.), and code efficiency. The system scores and feedbacks students' homework through AI algorithms, which can not only quickly judge whether the code can correctly realize the functions required by the topic, but also analyze the quality of the code. For syntax errors, logical errors, and non-standard parts in the code, the system will give specific feedback suggestions to help students understand the causes of errors and make improvements. For example, it points out that variable naming is not standardized and suggests using more descriptive names; prompts that the loop structure can be further optimized to improve code execution efficiency, etc. At the same time, the system should be able to integrate with common online learning systems as much as possible, facilitating teachers to use it on existing teaching

platforms, realizing a one-stop process of homework submission, correction, and feedback, and greatly improving teaching efficiency.

4.4 Adding Application Cases of Python in the AI Field

With the rapid development of artificial intelligence technology, Python is increasingly widely used in the AI field. In order to enable students to better understand the practical application value of Python and stimulate their learning interest, application cases of Python in the AI field should be added to the curriculum content.

After explaining Python basic syntax and data structures, simple data analysis cases can be introduced. Use Python's pandas library for data cleaning, processing, and analysis, such as analyzing a piece of student grade data, calculating the average, highest, and lowest scores of each subject, and counting the number distribution of each score segment, etc. Through practical cases, students can master the basic process and methods of data processing and understand the powerful functions of Python in data processing.

Furthermore, basic machine learning cases can be introduced. Use the scikit-learn library to implement simple classification and regression tasks, such as training and predicting classification models using the iris dataset. Through operations such as adjusting model parameters and evaluating model performance, students can initially understand the basic principles and processes of machine learning and the application methods of Python in machine learning. Deep learning cases can also be introduced, and simple neural network models, such as handwritten digit recognition models, can be built with the help of TensorFlow or PyTorch frameworks, allowing students to experience the charm of deep learning and feel the convenience of Python in building complex models. The introduction of these cases can enable students to closely combine Python programming knowledge with the popular AI field, improving their ability to apply knowledge and learning enthusiasm.

4.5 Designing Project-Based Learning Content Based on Real Scenarios

Project-based learning can enable students to apply the knowledge they have learned in actual projects, improving their ability to solve practical problems and teamwork ability. Therefore, designing project-based learning content based on real scenarios is an important direction for curriculum content reconstruction.

A project of "Data Analysis and Visualization of Small E-Commerce Platform" can be designed. In this project, students need to collect sales data of the e-commerce platform, which may include product information, order data, user reviews, etc. Then, use Python's relevant libraries, such as pandas for data cleaning and preprocessing, removing duplicate data, handling missing values, etc.; use numpy for numerical calculations, calculating key indicators such as sales volume, profit, and user purchase frequency; use matplotlib, seaborn and other libraries for data visualization, drawing product sales trend charts, user geographical distribution maps, pie charts of sales proportions of different product categories, etc. Through data analysis and visualization, students need to find patterns and problems in sales data and put forward corresponding marketing strategies and suggestions.

Another example is designing an "Intelligent Chatbot Development" project. Students first need to understand the basic principles and technologies of natural language processing, then use Python's NLTK (Natural Language Toolkit) or more advanced Transformer frameworks, such as relevant libraries of Hugging Face, to develop chatbots. They need to collect and sort out dialogue datasets, preprocess and annotate the data, train language models, and realize the basic functions of the chatbot, such as understanding user questions and generating appropriate answers. In the process of project implementation, students also need to continuously optimize the model performance to improve the accuracy and fluency of the chatbot. Through these project-based learning based on real scenarios, students can deeply master Python programming skills in practice and cultivates innovative thinking and practical work ability.

4.6 Construction and Application of Interactive Teaching Platform

Building an interactive teaching platform based on AI technology can provide students with a richer and more convenient learning experience. The platform integrates a real-time coding environment, an instant feedback system, and an intelligent tutoring function.

In the real-time coding environment, students can write Python code directly on the platform and run it to view the results immediately. Compared with the traditional local development environment, there is no need for tedious environment configuration, reducing the learning threshold. The platform's instant feedback system can detect errors in students' code in real time and give detailed error prompts and modification suggestions. When a student enters incorrect syntax, the system will immediately pop up a prompt box indicating the error location and type, such as "Syntax error: missing parenthesis here". The intelligent tutoring function uses AI technology to provide personalized guidance and answers according to students' questions and code conditions. When a student encounters difficulties in writing code to implement a complex function, they can ask the platform, and the intelligent tutoring system can analyze the student's existing code ideas and give targeted suggestions, such as prompting that a specific function of a certain library can be used to simplify the implementation process, or guiding the student to think from another angle to help them overcome programming obstacles.

Teachers can create course tasks, assign homework, and organize online tests on the platform. After students complete tasks and homework, the platform can automatically correct them and generate detailed learning reports, including information such as students' answer status, error analysis, and learning progress. By viewing the learning reports,

teachers can timely understand each student's learning status; find common and individual problems existing in students' learning process, so as to adjust teaching strategies targetedly, conduct individual tutoring or centralized explanation. The application of this interactive teaching platform can enhance students' learning participation and improve teaching effects.

4.7 Introducing AI Virtual Tutors to Realize 24/7 Learning Support

AI virtual tutors can serve as students' exclusive learning partners, providing all-round and round-the-clock learning support. Equipped with natural language processing capabilities, virtual tutors can understand various questions raised by students and respond in an easy-to-understand manner. Whether it is confusion about Python syntax or difficulties in grasping complex algorithm logic, students can turn to virtual tutors for help at any time. For instance, when students are learning the concept of Python decorators and struggle to understand their functions and usage, they can ask the virtual tutor, "What is a Python decorator and what is its purpose?" The virtual tutor can use vivid examples to explain—such as demonstrating how decorators add new functionalities (e.g., log recording or performance monitoring) to a function without modifying its original code to help students comprehend this relatively abstract concept.

Virtual tutors can also offer personalized learning suggestions based on students' learning progress and historical records. After students complete studying a certain knowledge point, the virtual tutor can recommend supplementary learning materials or practice questions according to their performance in related exercises. If a student frequently makes index errors in list operation exercises, the virtual tutor will suggest advanced practice questions on list indexing, slicing, and traversal, while also pushing video tutorials that explain common pitfalls in list operations, helping the student strengthen weak areas.

Furthermore, virtual tutors can simulate real programming scenarios and engage in code debugging dialogues with students, guiding them to independently identify and solve problems, thereby gradually cultivating their ability to program independently. They truly become a reliable, always-available assistant in students' Python learning journey. In simulating large-scale project development, virtual tutors can even act as team members, discussing code architecture and functional module design with students. For example, when a student is planning a Python web crawler project, the virtual tutor can propose suggestions from perspectives like anti-crawling strategy responses and data storage optimization, guiding the student to refine the project plan. This allows students to accumulate practical project development experience through virtual collaboration, laying a solid foundation for their future career development.

Throughout the long-term learning process, virtual tutors can continuously record and analyze students' learning data to generate personalized learning growth curves, clearly illustrating changes in students' mastery of various Python programming knowledge modules. Based on this data, virtual tutors can set phased learning goals for students and plan key areas for the next learning stage, helping them make steady progress in Python programming and transform from beginners to professional developers. For students with weaker practical abilities, virtual tutors can arrange progressive coding exercises and provide corresponding improvement examples, helping them enhance code quality and develop good programming habits. This truly achieves "teaching students in accordance with their aptitude," enabling every student to efficiently advance their Python programming skills with the companionship of an AI virtual tutor.

5 CONCLUSION

The teaching reform of Python programming course with AI empowerment is an important measure to cope with the limitations of traditional teaching mode and meet the needs of talent training in the digital age. There are many problems in traditional Python teaching in teaching mode, learning resources, practical teaching and course evaluation, such as low student participation, lack of personalized guidance, single resource form and lagging update, practice projects divorced from reality, and one-sided evaluation methods.

With the help of AI technology, by applying AI auxiliary tools such as intelligent programming assistant, personalized learning path planning tool and automatic homework correcting system, we can reconstruct the course content including application cases and real scene projects in the AI field, innovate teaching methods such as interactive teaching platform and AI virtual tutor, and simultaneously promote the transformation of teachers' role into "learning designers" and build a multi-dimensional intelligent evaluation system, which can effectively solve the pain points of traditional teaching.

This reform can not only enhance students' interest in programming, learning efficiency and ability to solve practical problems, but also make teaching more suitable for technical development and industry needs, and finally realize the transformation of Python programming education from "standardized knowledge transfer" to "personalized ability training", and inject new kinetic energy into cultivating high-quality programming talents adapted to the era of artificial intelligence. In the future, with the continuous development of AI technology, Python programming teaching will continue to deepen in personalization, scene and intelligence, forming a virtuous circle of "empowering teaching with AI and feeding back AI applications with teaching".

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

FUNDING

The author greatly appreciate the sponsorship from 2 teaching reform programs jg2024012 and ZNJY2410 from Zhejiang Shuren University.

REFERENCES

[1] Chen K, Zhang J, Xie X. Research on Integrating Political Ideology into Python Programming Course. Modern Business Trade Industry, 2023, 44(23): 227-229.
[2] Liang L. Teaching Reform of Python Programming Course for Non-computer Majors—Taking Financial Management Major as an Example, Modern Business Trade Industry, 2024, 45(14): 208-210.
[3] Lv Y, Wu Y. Research on Teaching Reform of Business Analytics Course in Digital Era with Big Data—Taking "Python Programming for Economics and Finance" as An Example. Big Data, 2025, 11(01): 46-55.
[4] Wei M. Reserch on Teaching Reform for Financial Data Analytics Course. Pivot, 2022, 10: 118-120.
[5] Ibrahim H. Curriculum Reform and Teacher Autonomy in Turkey: The Case of The History Teaching. International Journal of Instruction, 2011, 4(2): 113-128.
[6] Leigh K, Sherry A. Reforming Practice or Modifying Reforms?: Elementary Teachers' Response to the Tools of Reform. Journal of Research in Science Teaching, 2007, 44(3): 396-423.
[7] Wu X. Research on the Reform of Ideological and Political Teaching Evaluation Method of College English Course Based on "Online and Offline" Teaching, Journal of Higher Education Research, 2022, 3(1): 87-90.
[8] Moshe B, Larisa S. Reform-Based Science Teaching: Teachers' Instructional Practices and Conceptions, Eurasia Journal of Mathematics, Science & Technology Education, 2008, 4(1): 11-20.