Research on the Development and Practice of Programming Design Courses for Higher Vocational Finance and Economics Majors Based on Computational Thinking Cultivation -in case of Application of Python in Finance

Yi Peng[1], Yanan Chen[2*]

1.School of Economics, Shenzhen Polytechnic University, Shenzhen, 518055, China
2.School of Management, Guangzhou City University of Technology, Guangzhou, 510800, China
[*] Corresponding author: chenyanantina@163.com

Abstract: *This study takes the course Application of Python in Finance as a case example, utilizing a project-based teaching model to develop vocational finance program design courses that can effectively enhance students' computational thinking skills and programming abilities to meet the demands of future career development. The study employs a pre-test and post-test design to assess the students' progress in computational thinking and programming abilities before and after the course. The results show that the Application of Python in Finance course significantly enhanced students' programming abilities and multiple dimensions of computational thinking, including problem decomposition, pattern recognition, algorithm design and implementation, as well as testing, evaluation, and optimization.*

Keywords: *Computational Thinking; Python in Finance; Higher Vocational Financial Majors; Program Design Courses*

Introduction

In the digital age, computational thinking has become an indispensable core competency, encompassing aspects such as problem-solving, logical reasoning, abstract thinking, and algorithm design. For students in higher vocational financial majors, cultivating computational thinking is not only key to adapting to future career demands but also an important way to enhance their overall competence and innovation abilities.

Python, as an efficient and powerful programming language, is increasingly applied in the finance sector. Its concise syntax, robust data processing capabilities, and extensive library support make it an essential tool in financial analysis. The Application of Python in Finance course provides an effective avenue for financial majors to cultivate computational thinking. Through a project-based teaching model, students can deeply understand the concepts of computational thinking, such as decomposition, pattern recognition, and abstraction, while solving real-world problems. The course not only enhances students' computational thinking skills but also allows them to experience the role of computational thinking in solving complex problems in the real world. Moreover, Python's ease of use enables students to quickly master it, allowing them to focus more on developing and applying computational thinking skills.

1. Issues in the Python Programming Course for Higher Vocational Financial Majors

1.1 Unclear Course Objectives

In some higher vocational financial programs, there is an issue of unclear course objectives in Python programming courses. When offering Python courses, there is a lack of in-depth consideration regarding course objectives. The course content is not designed based on students' actual situations and needs, and there is a lack of clear understanding and planning regarding the specific abilities that the

course aims to develop. As a result, the course often overemphasizes the syntax and basic operations of the programming language, neglecting the application of Python in data analysis, problem-solving, and algorithm design, which are key components of computational thinking[1].

1.2 Lack of a Systematic Teaching Model

In higher vocational financial programs, Python programming courses still follow traditional teaching models. Traditional models tend to focus on teaching knowledge points, which leads to a lack of student initiative and engagement, diverging from Python's practical characteristics. Project-based teaching, which allows students to learn and apply knowledge while solving real-world problems, is not effectively implemented in many vocational colleges, resulting in students' inability to translate theoretical knowledge into practical skills.

2. Development of Python Programming Courses Based on Computational Thinking for Higher Vocational Financial Majors

Computational thinking is not only a skill exclusive to programming or computer science; it is an interdisciplinary way of thinking that is also applicable to the financial sector.

2.1 Teaching Development Basis

2.1.1 Characteristics of the Program Design Course for Higher Vocational Financial Majors

The Python programming course for higher vocational financial majors is characterized by interdisciplinary integration and strong practicality.

This course integrates knowledge from finance and information technology, facilitating interdisciplinary teaching. It covers not only basic skills such as programming languages and software development but also addresses practical issues in the financial sector, such as accounting and finance. The course aims to equip students with the ability to apply technology to solve financial problems. For example, students can use programming to conduct financial data analysis and risk assessments, allowing them to master technical skills while understanding and applying financial theory, thus achieving an organic combination of technology and financial knowledge.

The course emphasizes practical operations and skill development, closely aligned with the demands of the finance industry. The teaching content focuses on practical programming techniques, such as web scraping and data analysis, enabling students to learn problem-solving methods through project practice. The course design is centered on case-based teaching, simulating real work scenarios in the financial sector to enhance students' professional skills and employability. Practical teaching is emphasized in the course, with many hands-on sessions, project-based training, and other activities to ensure that students can consolidate and deepen theoretical knowledge through practical experience[2].

2.1.2 Computational Thinking Training Model

Computational thinking can be mastered through programming practice. In the Application of Python in Finance course, cultivating computational thinking should be a course objective and the basis for course development.

The problem-solving process of computational thinking is carried out step by step. The first step, problem definition and decomposition, requires identifying the specific content and objectives of financial problems and breaking them down into smaller, more specific subproblems. The second step, pattern recognition and abstraction, involves identifying patterns and trends in the data related to financial problems. Through abstraction, key financial indicators and data features are extracted, allowing students to focus on solving core issues. The third step, algorithm design and implementation, involves designing algorithms to solve the identified subproblems and implementing these algorithms as specific programs or tools to automate financial processes. The fourth step is testing, evaluation, and optimization; testing the effectiveness of algorithms and tools, collecting feedback, and debugging and optimizing based on test results. The accuracy, efficiency, and impact of the solution on financial processes should be evaluated, and algorithms should be further optimized based on these evaluations to improve overall processing quality and efficiency. During the course development, it is essential to integrate the steps of problem-solving through computational thinking into the teaching development process.

2.2 Principles of Instructional Design

The project-based teaching design for the Python in Finance course for higher vocational financial majors needs to follow the principles of fostering information awareness and integrating with the profession.

In financial education, fostering information awareness is crucial for students, encompassing sensitivity to information, judgment skills, and problem-solving abilities. The Python programming course, with its powerful data processing capabilities, especially through the use of libraries like Pandas, enhances students' sensitivity to data, enabling them to quickly identify valuable information from large volumes of data. At the same time, the course strengthens students' independent judgment regarding the reliability and timeliness of information, sparking their interest in artificial intelligence and emerging technologies. By exploring cutting-edge technologies like machine learning and data mining, students' curiosity and thirst for knowledge are heightened, motivating them to actively seek new methods for solving problems[3].

The course must be integrated with the financial profession. At the foundational level, it includes concepts such as coding standards, flow control, and data structures. By incorporating these concepts with finance-related case studies, students' understanding of the profession is enhanced, and their interest in learning is increased. At the advanced level, the course focuses on data analysis, machine learning, and other topics, with teaching content tailored to professional needs. For instance, in accounting, the focus would be on financial data analysis and web programming to achieve a complete workflow from data acquisition to processing.

2.3 Instructional Content Design

When designing the course, the teaching content must be carefully selected based on the characteristics of the profession and students' needs. Throughout the teaching process, emphasis should be placed on cultivating computational thinking.

Project 1: Unveiling Python. In this project, students will initially learn about Python from the perspective of computational thinking. The process starts with problem definition and decomposition, where the learning objectives are clearly defined and broken down into sub-tasks, such as mastering basic syntax, data structures, and simple programming logic. Students will then become familiar with common patterns in Python code. Following the teacher's guidance, students will design and implement algorithms to solve specific programming problems, such as creating a simple calculator program. Finally, students will test the program's accuracy, evaluate its efficiency, and make optimizations based on feedback[4].

Project 2: Managing Sales Orders. This project involves learning Python control flow statements while developing computational thinking to manage sales orders. The first step is to define the sales order management problem and break it down into sub-tasks, such as customer classification, discount calculation, and performance statistics. Next, students will recognize patterns in sales data and abstract key attributes. They will then design algorithms to automate sales order processing and implement these algorithms to optimize discount applications and performance statistics. Finally, students will test the sales management system, evaluate its performance, and optimize it based on feedback.

Project 3: Managing the Purchasing System. In the purchasing system management project, students will apply computational thinking to master Python data structures and use them in the management of the purchasing process. The project begins by breaking down the purchasing management problem into sub-tasks, such as supplier management, raw material tracking, and purchasing decisions. Students will identify patterns in supplier and raw material data and abstract key elements of the purchasing system. Then, they will design and implement algorithms to optimize purchasing decisions. In the final stage of the project, students will test the purchasing system and assess its cost-effectiveness to improve purchasing efficiency.

Project 4: Retrieving Financial Data. In this project, students will learn data scraping techniques to automate the process of downloading financial data from websites. They will first understand the goals of data retrieval and break it down into sub-tasks, such as web page analysis, data extraction, and data storage. Then, students will recognize the storage patterns of financial data on web pages and abstract the key steps of the data scraping process. Under the guidance of the teacher, students will design algorithms to automate the data scraping process and implement these algorithms. Finally, they will test the scraper program's accuracy and optimize it based on feedback[5].

Project 5: Analyzing Financial Data.In this project, students will learn data cleaning and analysis techniques to conduct in-depth analysis of financial data. The project begins by defining the goals of financial data analysis and breaking them down into sub-tasks, such as data cleaning, calculating financial indicators, and trend analysis. Students will identify patterns in financial data and abstract key financial indicators. They will then design and implement algorithms to process and analyze the financial data. Finally, students will test the accuracy of their analysis results.

3. Teaching Practice of the Python Programming Course for Higher Vocational Financial Majors Based on Computational Thinking

3.1 Teaching Practice Arrangement

The core objective of this teaching practice is to integrate computational thinking into the Python Programming course for higher vocational financial majors. This is achieved through a project-based teaching framework designed to enhance students' programming and problem-solving skills. The target group for this practice consists of second-year students from a higher vocational college majoring in finance. The course applies project-based teaching methods to deeply practice the four key steps of computational thinking: problem definition and decomposition, pattern recognition and abstraction, algorithm design and implementation, and testing, evaluation, and optimization. The course aims to cultivate students' computational thinking through specific programming projects, enabling them to systematically analyze problems in the financial domain and design effective programmatic solutions.

3.2 Teaching Practice Process

Before the Python in Finance course begins, students are given a pre-test to assess their basic programming skills and computational thinking abilities. This pre-test evaluates students' understanding of Python programming language and its application in finance, as well as their computational thinking skills prior to the course. The anonymous nature of the questionnaire encourages students to respond based on their actual level, ensuring data reliability. The teacher supervises the completion of the questionnaire in class and ensures the timely collection of the questionnaires, guaranteeing the integrity and accuracy of the data.

Once the pre-test is completed, the teaching phase begins. The instructional practice focuses on cultivating students' computational thinking, combining theory and practice through project-based learning methods. The implementation process is illustrated using Project 5: Analyzing Financial Data as a case study.

The teacher introduces the problem definition and decomposition steps of computational thinking through real-world cases. They demonstrate how uncleaned financial data can lead to analytical biases and decision-making errors, guiding students to recognize the necessity of data cleaning. Through discussions and Q&A sessions, students learn how to break down complex financial data analysis problems into smaller, more specific sub-problems, such as handling missing data, outliers, and duplicates[6].

Through case analysis and programming practice, students apply the theoretical knowledge of computational thinking in practical operations. Students decompose the data cleaning task into smaller steps and practice after each step, which corresponds to the algorithm design and implementation process. Each student group presents their data cleaning results, including before-and-after data comparisons and analysis, representing the evaluation and feedback phase in computational thinking. The teacher summarizes the teaching content and student outcomes, emphasizing the importance of data cleaning and encouraging students to engage in self-reflection, which aligns with the reflection and improvement phase of computational thinking.

In this teaching process, students not only learn Python programming skills but, more importantly, learn how to apply computational thinking to analyze and solve practical problems in financial data analysis.

After the Python in Finance course's teaching experiment, students are given a post-test to assess their computational thinking and programming abilities. The purpose of this assessment is to measure students' programming level and computational thinking skills after completing the course's practical experiments. By comparing the pre-test and post-test results, we can evaluate the effectiveness of the Python in Finance course in improving students' programming abilities and provide feedback and

suggestions for future teaching practices.

3.3 Analysis of Teaching Effectiveness

In the Python Programming course, students underwent ability tests before and after the practice to assess the impact of programming design on computational thinking abilities. Before the practice, a questionnaire was used to evaluate students' initial programming skills and computational thinking abilities, allowing us to understand their capabilities before the course began.

After a period of project-based teaching practice, during which students participated in programming projects related to the finance major, they improved their Python programming skills and computational thinking abilities. After the practice, a follow-up test was conducted to assess students' programming and computational thinking abilities. The results showed a significant improvement in both students' programming skills and computational thinking abilities. Through the project-based Python course, students not only enhanced their programming skills but also made progress in computational thinking. They were better able to understand problems, design solutions, implement algorithms, and evaluate and optimize results.

4. Conclusion and Outlook

This research, based on the Python in Finance course, explores the development and practice of a programming course integrating computational thinking into the higher vocational finance curriculum. By adopting a project-based teaching model, this course aims to cultivate students' programming abilities and computational thinking skills, meeting the new demands of the digital age for finance professionals. The results of the teaching practice show that students' programming abilities and computational thinking skills were significantly enhanced after participating in finance-related programming projects.

Fund Project

Guangdong Province Quality Engineering Vocational Education Teaching Reform Research and Practice Project "Research on the Development and Practice of Programming Design Courses for Higher Vocational Finance and Economics Majors Based on Computational Thinking Cultivation in case of Application of Python in Finance" in 2023(2023JG333).

Guangdong Education Science Planning Project (Higher Education Special Project) (2024) "Research on Coordination Mechanism and Optimization Path of New Productivity Development in Guangdong Higher Vocational Education Empowering Areas" (2024GXJK771)

References

[1] Yi Peng and Yanan Chen and Bowen Li. Teaching Reform of "Python Programming" for Accounting in the Context of Artificial Intelligence —Discussion on Project Driven Teaching Mode[J]. Advances in Higher Education, 2021, 5(2)

[2] Zhang, X., Liu, X. G., & Li, M. R. (2022). Exploration and practice of Python Programming course teaching under the new financial background. Industry and Technology Forum, 21(11), 157-158.

[3] Yan, J. (2023). Path for the construction of the "Python Programming" course under the perspective of professional digital upgrading: A case study of a financial higher vocational college in Changchun. Science and Technology Information, 21(07), 123-127.

[4] Xiao, Q., Huang, W. L. (2023). Exploration of improving college students' data literacy under the new liberal arts background: Embedding Python language and data analysis in a public course. University Education, (11), 40-43.

[5] Ge, X. J., Zhao, J. B., Ping, R., et al. (2024). Teaching research and practice of non-computer major programming courses based on computational thinking. Chinese Journal of Information Technology Education, (17), 109-112.

[6] Liao, J. G., Liang, W., Han, X., et al. (2024). Project-driven teaching research with practical application: A case study of Python programming course for financial management majors. Higher Education Journal, 10(04), 58-61+66.