

Career-oriented Python Course Design Based on the Background of New Engineering

Chaofeng Wang and Qiao Xiao

School of Computer Science, University of South China, Hengyang 421001, China

Abstract

Based on the educational background of the new engineering and the concept of career orientation, this work investigates how to design a Python course that meets the high demands of cultivating engineering talents who would perform a crucial role in the development of the society. From the perspectives of teaching objectives, teaching design, teaching methods, and teaching evaluations, this work presents the detailed practical experiences of designing a career-oriented python curriculum. Through exploration and practice, this work proposes some teaching models and methods with a goal to provide guidance for the reform and teaching of Python courses. The results show that, with the proposed course design, the quality of education is improved and students' comprehensive abilities are enhanced, which demonstrate the efficacy of the proposed design.

Keywords

Python course construction; New engineer; Career-oriented; Teaching design.

1. Introduction

In recent years, the rapid development in the field of computer science and technology has presented new challenges to computer education [1]. As an advanced programming language widely applied in software development, data analysis, and artificial intelligence, Python plays a significant role in engineering practices. To cultivate students' programming skills and problem-solving abilities, an increasing number of educational institutions have incorporated Python programming into their curriculum. Based on the existing research and practice, traditional designs of Python course are relatively mature and have been widely implemented in many educational institutions and universities [2]. The basic idea of the traditional designs primarily focus on establishing foundational knowledge of the Python programming language and developing programming skills under the Python frameworks [3]. The following are some common characteristics of traditional Python course offerings from various perspective: 1) The objectives of traditional Python courses are intended to motivate students to comprehend and apply fundamental concepts and techniques of the Python programming language. Typically, course objectives include familiarizing students with Python syntax, mastering basic data structures and algorithms, and the ability to write simple programs. Those objectives aim to cultivate students' programming mindset and problem-solving abilities. 2) The teaching content of traditional Python courses primarily encompasses fundamental knowledge of the Python language, data types, control flow, functions, file operations, and more. The course content is usually organized in a progressive manner, starting from foundational knowledge and gradually advancing to higher-level applications. Textbooks and teaching resources are frequently employed to support student learning. 3) Traditional Python courses often employ a combination of traditional classroom lectures and practical exercises. Teachers guide students' learning through methods such as concept explanations, demonstration of examples, problem-solving, and programming exercises. Some courses may also incorporate laboratory activities and programming projects to reinforce students' practical skills. 4) The assessment

of traditional Python courses typically relies on students' assignments, exams, and project performance. Assignments may involve programming exercises, small projects, or problem-solving tasks. Exams are utilized to evaluate students' understanding of Python syntax and concepts. Additionally, some courses may consider students' classroom participation and contributions. However, the design of Python programming courses needs to focus on students' practical needs and career orientation in order to adapt to the rapidly changing industry demands. Career-oriented Python courses have gained increasing attention in the field of education to meet the career development needs of students [4,5]. These courses aim to cultivate students' practical application abilities, enabling them to flexibly apply Python programming skills in professional settings. Here are some common characteristics of career-oriented Python course offerings [6,7]: 1) The objectives of career-oriented Python courses are to equip students with practical Python programming skills relevant to real-world work scenarios and enable them to apply these skills in various professional domains. Course objectives typically include students mastering the use of Python libraries and frameworks, understanding software development processes, engaging in project development, and practical applications such as data analysis and machine learning. 2) Career-oriented Python courses often base their content design on real work scenarios and projects. In addition to foundational knowledge of the Python language, the course content includes topics relevant to professions, such as web development, data analysis, artificial intelligence, and more. Teaching resources and case studies are frequently used to support student learning and practical application. 3) Career-oriented Python courses typically adopt project-driven and practice-oriented teaching methods. Teachers guide students through practical activities that involve real project development, problem-solving, and facing professional challenges, helping students enhance their practical application abilities and teamwork skills. Students may also have opportunities to engage with real work environments through internships and practical training. 4) The evaluation of career-oriented Python courses often focuses on students' project outcomes, practical application abilities, and teamwork skills. Project assessment, code quality, problem-solving capabilities, and students' roles and contributions within a team are considered important indicators for evaluation. By incorporating career-oriented Python courses, students could develop their programming skills comprehensively and prepare for their future career development. Such course designs combine academic theories with practical applications, enabling students to flexibly apply Python programming skills in real-life scenarios, better meeting industry demands and career development requirements [8].

In this context, this work aims to explore career-oriented Python course design, providing an effective teaching model that promotes students' career development and practical skills enhancement. Specifically, this article will extensively discuss the design of career-oriented Python courses, covering aspects such as setting course objectives, designing teaching content, selecting teaching methods, and establishing an evaluation system. Through comprehensive analysis of relevant literature and summarization of practical teaching experiences, a systematic and practical course design framework will be provided to assist educators in organizing and implementing Python programming instruction more effectively.

2. Development of Career-oriented Python Course

When designing a career-oriented Python course, the focus should be on cultivating students' ability to apply Python programming skills in professional fields. Here are some teaching measures taken to design a career-oriented Python course.

1) Teaching Objectives: Clearly define the goals and learning outcomes of the course. For example, to let students master the related topics and abilities for the current status of artificial intelligence, the Python libraries such as Numpy, Pandas, Seaborn for data analysis are

extensively studied while two deep learning frameworks, i.e., TensorFlow and Pytorch are introduced. In addition the libraries for data visualization should be also included in this course. To improve the students' abilities to solve career-related problems, those problems will be extensively studied in this course. Moreover, the practical projects which requires real-world problem solving techniques are also included in this course. In summary, the objectives of this course is highly aligned with industry demands, which could in essence help improve the professional skills of students when they are facing real problems.

2) Curriculum Design: When designing the course content, incorporate the requirements of professional practice. Choose topics and case studies that are relevant to careers. The topics are diverse as the applications of Python language is wide such as web development, data analysis, machine learning, etc. Also, according to students of different majors, the curriculum designs would be different. For example, for students majoring in software engineering, the applications of python for web development or development in PC are interested to them. For students majoring in big data, the data analysis with statistical methods are the requirement for their graduation. For students in computer science, the implementation of machine learning algorithms with Python language is the emphasis of their study, Hence, the course content should be closely connected to real-world applications and covers a comprehensive range from fundamental knowledge to advanced applications but also meet the requirements for students in different subjects.

3) Practical Projects and Case Studies: Introduce practical projects and case studies that allow students to apply their Python programming skills in real work scenarios. These projects can be simulated or real-life collaborations with the industry, aimed at cultivating students' practical development experience and problem-solving abilities. For example, for application development, a popular web applications for data mining can be a case study. For data analysis, the statistical design and inference for medical data and social data can be exploited as a course project. For machine learning, the movement detection and face classification can be leverage as practical projects. Hence, based on those projects and case studies of practical values, the students could development their own interests to motivate them for further investigations.

4) Use Industry-Standard Tools and Frameworks: Teach students how to use commonly used Python tools and frameworks in the industry, such as Django, NumPy, Pandas, etc. Python has many mature packages and frameworks that provide developers with a wide range of functionalities and tools, greatly simplifying the development process. For example, Pandas is a library for data manipulation and analysis. It offers flexible data structures and data manipulation tools. TensorFlow and PyTorch are popular deep learning frameworks. They provide high-level interfaces for building and training neural networks, supporting the implementation and optimization of various deep learning models. They are widely used in fields such as image recognition, natural language processing, and recommendation systems. These mature packages and frameworks significantly enhance the functionality and efficiency of Python, allowing developers to develop various applications more quickly and efficiently. They are essential for data processing, scientific computing, machine learning, and many other fields. Familiarize students with the usage of these tools and their applications in professional fields.

5) Emphasize Practical Skill Development: Focus on developing students' practical application abilities. For example, involve students in different stages of the software development process, including requirement analysis, system design, coding, and testing. Through practical skill development, students can enhance their programming abilities and teamwork skills through project work. The traditional Python course often focus on "right" or "wrong" of their study, i.e., the results is more important than the process of the study. However, in practical, the answers to any problem is not fixed rather than having many possibility. Hence, the practical skill

development should be emphasized to motivate students to seek for the process to reach the goal.

6) Industry Mentors and Guest Lecturers: Invite industry mentors and guest lecturers to participate in the teaching of the course. They can share their practical experiences and the latest industry trends. They can provide industry case studies, guide student project development, and offer guidance on employment and career development.

7) Establish an Evaluation System: Establish a diverse evaluation system, including project assessments, code quality, problem-solving abilities, and teamwork skills. The evaluation should accurately reflect students' abilities and performances in practical applications.

8) Continuous Updates and Industry Alignment: Continuously update the course content and case studies to align with the ever-changing industry trends and developments. Stay connected with industry professionals to ensure the course remains relevant and up-to-date.

3. Results

The proposed course is practiced for students majoring in two subjects, i.e., big data and software engineering, including a total of 201 students. A rough statistical analysis is performed through discussion and evaluation from those students. Based on the statistical results, we can conclude that:

1) Most students think they have significant improvement in practical application skills. Through participating in practical projects and case studies, students can apply Python programming skills in real work scenarios, thereby enhancing their practical application abilities.

2) Most students think their teamwork and practical skills are enhanced. Through practical projects and case studies, students can collaborate with other students to complete projects, fostering their teamwork and communication abilities.

4. Conclusions

The demand for Python courses in the new era is characterized by diversified application domains, practice-oriented teaching, interdisciplinary integration, data-driven instruction, and flexible learning approaches. These characteristics reflect the requirements of the era and the trends in technology, providing students with a more comprehensive and practical Python programming education. Though the career-oriented Python course, students can gain practical programming experience and application skills, better meet the demands of career development, and enhance their competitiveness in the job market. Additionally, we will make adjustments and improvements based on the experiment results to further enhance students' learning outcomes and professional abilities.

Acknowledgments

This study is supported by the Teaching Reform Research Project of University of South China (2021YB-XJG31, 212JXJ046, 2021YB-XJG15).

References

- [1] J. Smith: The importance of Python programming in computer science education. *Journal of Computer Science Education*, 2019, 25(2), 123-145.
- [2] B. Li: Exploration of case studies in python programming design. *Digital Techniques and Applications*, 2023, 41(3): 8-13. (In Chinese)

- [3] Z. Feng, P. He, F. Li, et al.: Investigation on case-driven “ mastering theory+focusing on practice” conventional Python course. *Computer Education*, 2023, 339(03): 199-202. (In Chinese)
- [4] L. Zhang, Y. Zhang, and Y. Sun: Research on mixing teaching mode for Python course under the background of new engineering background. *Computer Era*, 2023, 368(02): 125-127. (In Chinese)
- [5] T. Wang, C. Tian, and J. He: Investigation on teaching design in Python programming under the background of innovation and entrepreneurship. *Innovation and Entrepreneurship Theory, Research, and Practice*, 2023, 6(03), 24-26. (In Chinese)
- [6] A. Brown and M. Johnson: Incorporating industry-oriented Python programming in computer science curricula. *International Journal of Computer Science Education*, 2020, 12(3), 215-230.
- [7] L. Wang and S. Chen: Developing a career-oriented Python programming course for software engineering education. *Journal of Software Engineering Education and Development*, 2018, 10(1), 67-84.
- [8] L. Liu, X. Gao: Exploration of teaching reform in vocational education of Python language programming based on outcome-oriented approach. *Computer Knowledge and Technique*, 2023, 19(01), 160-162. (In Chinese)