# Research and Practice of Case-Driven Teaching in Python Language Programming

Huiwen Zou[1], Xuebin Hong[1], Weiwei Zhao[1], and Tian Cai[2(✉)]

[1] Department of Computer, Shantou Polytechnic, Shantou 515078, China
{xbhong1124,wwzhao}@stpt.edu.cn
[2] Skill Training Center, Shantou Polytechnic, Shantou 515078, China
tcai45@126.com

**Abstract.** Python Language Programming acts significantly in computer teaching to meet the increasing market demand. However, the object-oriented characteristics of python language makes it difficult for students to learn and master, and the traditional teaching mode cannot effectively ensure the teaching effect. In this paper, Plan-Do-Check-Act (PDCA) Cycle, which was initially taken as a problem-solving measure in business, was introduced into Python language teaching as a case-driven teaching method. A case study was carried out to on computer students at Shantou Polytechnics. The case-driven teaching approach improve the quality of theoretical teaching, more effectively provide practical operation ability and explore effective teaching.

**Keywords:** Python educating · Programming ability · Case driven teaching · PDCA

## 1 Introduction

Python is one of the most popular tools which ranked first on TIOBE in 2023 [1, 2]. In 2016, the Ministry of Education's College Computer Course Teaching Steering Committee suggested Python language should be a teaching language for introductory programming courses [3]. In 2017, the Ministry of Education Examination Center added the course Python Language Programming to train students participating in management or computer applications in enterprises and institutions [4]. The current teaching method of "teaching + practice" leads to a shortage of innovation and progress in computer experiment teaching. In addition to theoretical consolidation in theoretical learning, programming ability requires practice and application to gradually develop [10].

In 1950s, Feigenbaum and expert JuLan of the General Electric Company of the United States proposed the concept of Total Quality Management [7] and involved to PDCA cycle by Dr. Deming. Four steps of this cycle are as the basis, namely Plan (plan), Do (execution), Check (check) and Act (processing). In the teaching quality management activities, various tasks are divided into planning, plan implementation, and inspection of implementation effects, and then the successful are included in the standard, and the rest are left for the next cycle to solve top form a closed-loop structure. These four stages are

also a very effective tool during teaching time. This is also the basic method of quality management, and the general law of various work in enterprise management [8]. In this paper, PDCA Cycle was introduced into Python language teaching as a case-driven teaching method and a case study was applied for students of computer department at Shantou Polytechnics.

## 2  The Case -Driven Teaching in Python Language Programming

The case-driven teaching method is to start with practical examples, and then rise to the theoretical level, and properly explain theoretical knowledge or corresponding knowledge points. The characteristic is from cases to theories or knowledge points, from sensibility to rationality. Case teaching refers to teaching and covers learning. In the teaching process, teachers create problem situations and take specific meaningful and specific tasks that students are interested in as clues, and subtly imply knowledge points in each task. Under the predetermined situation, students solve problems after thinking and with the guidance of teachers and acquire knowledge and cultivate professional ability through independent construction [5].

Take the specific needs of enterprise interviews and work as supplementary materials for teaching. Communication with companies involved in PYTHON language development and university teachers of related subjects before launching teaching courses to formulate more effective teaching methods, teaching content and teaching objectives. Python case teaching method attracts interest with graphics [6] and gives an introductory case and related graphics cases based on the turtle library that are set up to cooperate with the development of each teaching content.

The case-driven teaching method in "Python Language Programming" is given in Table 1. The above four steps do not end after running once, but loop. After each cycle, some pure problems in teaching will be solved, and the unsolved problems will enter the next cycle, so that the effect of circling is rising, as shown in Fig. 1.

**Table 1.**  PDCA implementation steps and content

| STEP | CONTENT |
| --- | --- |
| P (Plan) | 1. Determination of goals<br>2. The specific formulation of the content of teaching activities |
| D (Do) | 1. Design specific methods and programs<br>2. Carry out specific operations according to the design and layout<br>3. Realize what is planned |
| C (Check) | 1. Summarize the results of the execution plan<br>2. Distinguish the right and the wrong<br>3. Clarify the effect and find out the problem |
| A (Act) | 1. Process the results and affirm the effective experience<br>2. Summarize the lessons of failure<br>3. Submit the unsolved problems to the next PDCA cycle to solve |

**Table 2.** Program Design Scoring Rules

| ITEM | SCORE DETAILS | POINTS |
|---|---|---|
| Interface | 1. The interface is simple, generous, and friendly<br>2. Functional characteristics of the software<br>3. Display of the main functions included in the interface | 10 |
| Originality | 1.Degree of originality | 10 |
| Practicality | 1. Usability and maneuverability<br>2. Maintainability | 25 |
| Technical content | 1. Difficulty coefficient<br>2. The complexity of implementing functions<br>3. Better techniques and algorithms were used | 35 |
| Running effect | 1. Can run normally, Run without error prompt<br>2. Easy to operate | 10 |
| Teamwork | 1. Participation of each group member<br>2. Program division ratio | 10 |
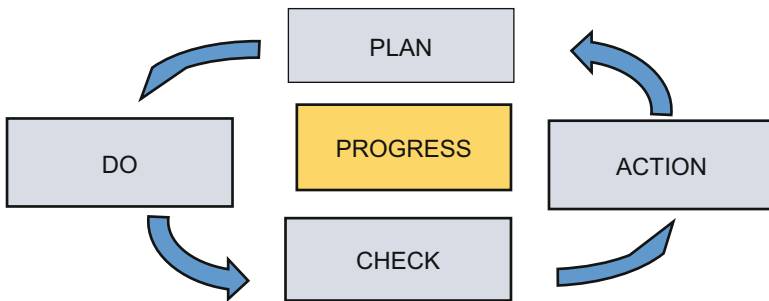


**Fig. 1.** Step by step PDCA model

## 3 Case Study

A case study was carried out to on computer students at Shantou Polytechnics. The teaching model proceeds the PDCA step by step.

### 3.1 Plan

The goal is to cultivate and improve students' programming thinking, and to assist the learning of other courses. Experiments and case-driven teaching are one of the important ways for students to understand the purpose of the course. Now the popular application directions, from Deep Learning, Web Scraping, Big Data, to traditional software development, all involve knowledge of the Python language. Therefore, how to make the "Python Language Programming" course more effectively serve these directions and at the same time be close to actual cases is what the explored in this paper.

At the same time, grade the homework of the students according to the program design scoring rules, as shown in Table 2. The grading rules for this program are written by the students themselves and then reviewed and determined by the teacher. In this way, after the students are grouped, they will comprehensively consider the program design according to the high requirements of the scoring rules.

Organizing inter-professional programming competitions and displaying excellent programming works in different classes has greatly improved students' enthusiasm and creativity in learning.

## 3.2   Do

Here, the OS module teaching in "Python Programming Language Design" is taken as an example. In addition to completing the basic examples in the textbook, combined with some basic customer needs learned during the company visits, some actual work project requirements are added as the example teaching part of the course. Such as the operation of batch processing file names in the computer file directory which is the main function in Bulk Rename Utility, a very practical file renaming tool, as shown in Fig. 2. During the teaching process, students are encouraged to analyze this software as the target to study the corresponding functions, and maintain an innovative attitude to develop more ideas and solutions.

The following is a brief design idea by a student.

1. The first step is to set the path to store the path (absolute path) entered by the user and define Suffix to store the files with the same suffix in the current directory that the user wants to modify before defining the name to store the new one that the user wants to modify filename content.
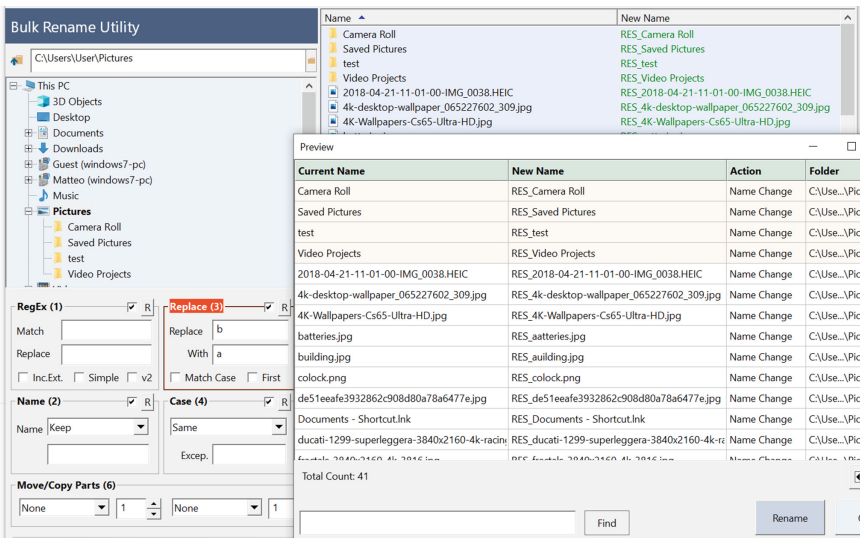


**Fig. 2.**  Partial interface of batch renaming in Bulk Rename Utility

2. The second step is to traverse all the files under the path value entered by the user, and then judge whether there is a match with the Suffix value entered by the user before calling the method. The content of the method is the same as the content entered by the user, and then automatically append i (i from 0, auto-increment), then append the Suffix value, and then rename.
3. The third step is to perform renaming when judging, and it will be displayed later that the old file name has been renamed to name $+$ _ i $+$ Suffix.
4. The fourth part, judge false, output directly, the old file name $+$ file does not belong to the $+$ Suffix format, and the renaming fails.

The following is a partial display of the code implementation among many students.

```
def main ():
        while True:
            print("1.Modify the file name 2.Check the File name 3.Exit")
            ans = int(input(' Please enter the relevant number to perform the oper-
    ation'))
            # Omit the judgment input statement
def xiugai():
        getpath = input(' Please enter the file path:')
        getname = input(' Please enter the file name you want to change:')
        houzhui = input('Please enter suffix name e.g(.txt)： ')
        path = os.chdir(getpath)
        list = os.listdir(path)
        for i in range(len(list)):
            if list[i].endswith(houzhui):
                os.rename(list[i],f'{getname}{i}{houzhui}')
        print(' Modification completed!!!')
def chakan():
        getpath = input(' Enter file path:')
        list = os.listdir(getpath)
        print ([i for i in list])
```

### 3.3 Check

Sufficient planning and preparation can ensure the smooth progress of the teaching process, and all the students in the class got excellent grades at the end of the term. This can be added to current teaching materials, as provides a valuable reference for future courses and related programming courses.

In this process, the students showed great interest to learn and develop a sense of competition, and the final academic performance in score of 18 weeks' generally improved compared to the past, as showed in Fig. 3. To avoid excessive competition, the form of class regrouping was carried out several times in different case projects. However, it is found that students with weak foundations will easily lose confidence in this relatively competitive environment, and some students plagiarized. Therefore,
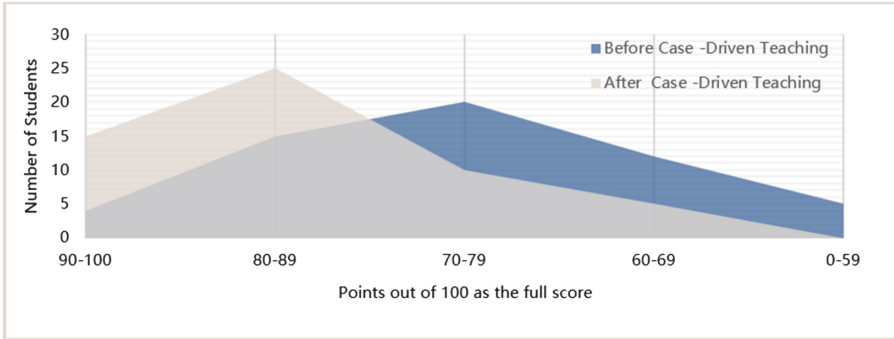
**Fig. 3.** Score Comparison Analysis before and after the Case -Driven Teaching

cultivate positive attitudes in normal teaching, and strengthen the education of ideals and beliefs in a subtle way [9].

### 3.4 Action

Check and inspect the details of the previous steps, and process the results of the summary inspection, affirm the effective experience, and summarize the lessons of failure to attract attention. Unresolved issues should be submitted to the next PDCA cycle for resolution. The spiral iterative of the project development cycle is developed to improve the practicability of the teaching model in a continuous cycle. It can efficiently and deeply cultivate students' ability to deal with real scenarios of computer programming.

## 4   Conclusion

The case-driven Python teaching method adopts case elements combined with classroom content to stimulate students' enthusiasm for learning and improve their practical ability. For future teaching, there are mainly three aspects to carry out. The first is to adopt the PCDA cycle to improve the teaching quality and teaching mode step by step, and explore and solve the problems in teaching in the cycle. In addition, combined with the latest market demand and the continuous update of Python language applications, case teaching materials are added and teaching methods are improved. Finally, to cultivate students' effective way of thinking in program design and methods of analyzing and solving problems, and to effectively try the successful case-driven teaching in other subjects.

## References

1. Victory House II, sp.401,5633 AJ Eindhoven, The Netherlands. The Software Quality Company, 022.Web. Mar2023. https://www.tiobe.com/tiobe-index/
2. Huang Haitao, Python3 artificial intelligence from entry to actual combat, People's Postal Press, May 2019.

3. Ministry of Education College Computer Course Teaching Steering Committee. Basic Requirements for University Computer Basic Course Teaching [M]. Beijing: Higher Education Press 2017.
4. Lin Zhe, Zheng Lin, Cai Tian. Exploration of Task-Driven Teaching Reform in Higher Vocational "Java Programming" Course [J]. Southern Vocational Education Journal. 2012.
5. The connotation of task-driven teaching method [J]. China Audio-visual Education, 2002 (9)
6. Huang Tianyu, Song Tian. Python case teaching method and practice of attracting interest with graphics [J]. Computer Education, 2017 (8): 32-37.
7. Jiang Hanwen. Research on Quality Management of Project A Based on Cost Control [D]. South China University of Technology, 2012.
8. Wan Rong. Introduction to Character Studies: Renmin University Press, 2013.
9. Gao Deyi, Zong Aidong. From Ideological and Political Courses to Course Ideological and Political Courses: Constructing a Curriculum System of Ideological and Political Education in Colleges and Universities from a Strategic Height [J]. Chinese Higher Education, 2017(1): 43-46.
10. Zhao Guanghui. Cross-integration case teaching of Python programming for new engineering [J]. Computer Education, 2017 (8) 23-27.