

Teaching Research on Object-Oriented Programming Language

Nengfa Hu*

Department of Computer Science and Engineering, Hanshan Normal University, Chaozhou 521041, China

Original Research Article

***Corresponding author**

Nengfa Hu

Article History

Received: 25.03.2018

Accepted: 07.03.2018

Published: 30.03.2018

DOI:

10.36347/sjet.2018.v06i03.001



Abstract: Aiming at the features of object-oriented programming (OOP) language and teaching objectives, a teaching method of OOP language based on case-based and task-driven teaching is proposed to further analyze its influence on teaching in terms of examination forms and propositional modes.

Keywords: object-oriented programming; task-driven; C# language.

INTRODUCTION

Object-oriented programming (OOP) language C# is a specialized compulsory course in software engineering major, which is a simple programming language with multiple functions. It combines the advantages of various languages including C++ and JAVA, showing the characteristics: simplicity, flexibility and being powerful. By taking succession as a mechanism, the language shares the methods and attributes of classification. C# language provides the delegation or agent mechanism to effectively realize the polymorphism of classification with an aim to reusing codes to further improve the development efficiency of software. The most prominent advantage of the C# language is to rapidly and efficiently develop various software based on network database by taking software engineering as a standard. Therefore, C# language has been well-received among numerous software engineers although it has been developed for ten years.

C# language exhibits numerous merits. However, how to make student recognize such merits and arise their real interests in the course is the key to teaching.

The author conducts establishes some educational reform pilots based on teaching practices for many years and test results has validated the ideal teaching performance.

Problems about teaching and suggestions

Many courses in universities have to be made with corresponding teaching objectives, requirements and evaluation modes. However, teachers tend to be in a difficult dilemma during teaching: the contradiction between cultivation of students' abilities and evaluation of test scores. This is mainly because numerous universities highlights the exam oriented teaching objectives while the test modes sometimes contradict with the cultivation of students' abilities, which is quite obvious during teaching of C# language. For example, many universities have stipulated that six topics at least are needed in a test paper, including multiple-choice quizzes and the exams, fill in the blanks, true or false items, and short-answer exam question, which is difficult for C# language teaching. The main reason is that teachers have to spare their time to conduct detailed teaching (including grammar, variables, constants and statements) under the guidance of tests. However, cultivating an ability requires that teachers and students should spend excessive time on details, because the key of mastering the course for students is to enable them

designing software with favorable performances using the knowledge learnt from the course. Students can preliminarily comprehend some detailed knowledge during software programming even though they do not mater the details, which hardly influence whole learning. However, test modes exactly waste too much students' time and energy, which is unfavorable for the cultivation of students' abilities. Current test systems of universities present a major defect, and therefore it is necessary to make reforms on test modes.

- Test modes should be diversified rather than only using one method. Universities should authorize teachers more initiative rights in test modes and strengthen the test monitoring.
- There are variable forms of test propositions. Different courses require diverse teaching objectives, requirements and modes so that diversified forms of test propositions are necessary.
- Teaching methods

The basic idea of the teaching C# language is summarized as follows: students can learn features of C# language and study from three aspects (basic knowledge of programming language, primary and advanced features of C# language) in view of theory

first and then practice. By doing so, the object-oriented idea and the new features of C# language are fully highlighted. By taking Windows application program (APP) and Web application development and data access of .NET platform as the principle line, the case-based and task-driven teachings are conducted, having achieved an ideal teaching performances, which are verified as an optimal method for students to learn the course [1,2].

Case-based methods

Case-based teaching refers to provide an environment of case study for learners and enable learners solving the problems and drawing feasible conclusions by collecting and analyzing data and making decisions. It aims to make students actively explore and participate in activities, improve their capabilities in cooperative communication and solving problems, which is a method by combining teaching and group discussion. Compared with class teaching, the method can more effectively combine the knowledge and practical application, inspire students' curiosity for knowledge and broaden the depth and width of students' knowledge. Thus, problems can be solved through innovative thoughts [3].

Case design for course teaching of C# language

Case teaching emphasizes the student-based ideas and cultivates the students' ability to solve practical problems by combining theories and practices. Therefore, quality of case design directly influences the effect of the teaching mode. Teachers should abide by the following principles while designing the cases of course teaching of C# language.

Objective

As C# language is integrated into .NET platform, which concentrates on Window APP, Web application development and network data access (core contents). Therefore, cases are chosen and designed based on the core contents.

Authenticity

By taking practical projects or chosen projects of enterprises and public institutions as research focuses, a simulation environment is created for students in order that students can get in touch with the practical software problems and environments in classes. It is in favor of improving the students' practical ability, proving stable theoretical and practical bases for students to be engaged in software development in the future. Moreover, it is necessary that universities invite experienced teachers and senior software engineers from some software companies to compile cases and demonstrate their project experiences and practical works developed based on C# language. The author designs a teaching website based on C# language in order to make students experience real development environment in teaching process. As the website

integrates SQL SERVER2000 server and FTP server of WEB server and students' works and solutions are processed using the APP. By doing so, students can truly get to know the powerful functions of C# language and their learning enthusiasm can be further inspired on the course.

Experimental characteristics

Given the method of software engineering development is fully applied in the process of developing C# language and it is necessary to focus on the practices during teaching. However, students are difficult to finish the experimental contents according to teachers' requirements in the experimental classes. To solve the problem, the best method is to fully consider the representativeness, typicality and operability of experimental cases which is conducive to leading students to complete the experimental tasks successfully. As a result, students not only can fully understand the theoretical knowledge and systematic theory but also can cultivate the ability to correctly solve problems and improve self-study ability under the constant enlightenment.

Design of task-driven teaching method in C# language teaching

The practical steps of task-driven teaching method in C# language teaching are shown as follows:

(1) After carefully designing tasks, the presentation on teachers' works with specific tasks is suggested to be made by primarily demonstrating the general framework of concrete steps for completing tasks. The core of C# language course is to cultivate the ability to develop software. However, it is required that students fully master multiple knowledge involving class mechanism of C# language, event and agents, ADO control and abnormal processing if they complete software development. Since such knowledge is difficult to be learned to some extent, so that teachers should consider this regard while designing tasks. Teachers are encouraged to select tasks which are easy to be understood and give proper hints. Therefore, the tasks designed by teachers should be interesting and curious and can impact students from the perspective of visual and aural senses so as to further inspire students' interests on learning knowledge and to help them build confidence of overcoming difficulties.

(2) Teachers should clearly assign tasks and the experimental tasks of C# language have to be specific and targeted. Students are encouraged to fulfill tasks or design new tasks so as to further master knowledge related to C# language.

(3) Preliminary completion of students on specific tasks requires timely feedback to teachers. Afterwards, teachers should compare and analyze the preliminary achievements that students have achieved, and then timely assign corresponding students the tasks which

are urgent to be completed but haven't mastered after dividing into small specific tasks. Students further explore the specific tasks and compare with the learned cases in teaching to find some rules.

(4) Students exchange with each other and evaluate their works, clearly point out and record the general and individual problems and independently complete new tasks proposed aiming at individual problems. Teachers have to put forward new specific tasks aiming at the general problems and then repeat (2) and (3) aforementioned according to progress of tasks' fulfillment.

(5) Finally, teachers need to come to a summary and make evaluation on the tasks completed and designed by students and select excellent works to lead students to have exchange in order to inspire students' learning in next stage.

CONCLUSIONS

Although C# language shows a common feature with all programming languages, it has characteristics which are required to be sufficiently considered in teaching. The purpose of learning C# language is to improve students' ability to develop software, which can lay a solid foundation for being engaging in software development in the future. Therefore, teachers should conduct teaching focusing on cultivating students' ability to develop software.

REFERENCES

1. Xiangyun Du, Ke Zhao, Youjin Ruan, Li Wang, Xiaoju Duan. Beginner CFL learners' perceptions of language difficulty in a task-based teaching and learning (TBTL) environment in Denmark. *System*, 2017,108-120.
2. Mohammad Bagher Shabani, Arad Ghasemi. The Effect of Task-based Language Teaching (TBLT) and Content-based Language Teaching (CBLT) on the Iranian Intermediate ESP Learners Reading Comprehension. *Procedia-Social and Behavioral Sciences*, 2014,1713-1721.
3. Marta Arrue, Silvia Caballero. Teaching skills to resolve conflicts with acute confessional syndrome patients in nursing using the Case Method (CM). *Nurse Education Today*, 2015,159-164.