Exploration and Practice of the Teaching Mode of C Language Programming Course for Cultivating Computational Thinking Ability

Gao Feng

Xi'an Technology and Business College, Xi'an, Shaanxi Province, China

Keywords: Computational thinking, Teaching models, Programming design, Competency development

Abstract: The traditional C language programming course teaching has the problem of emphasizing knowledge transfer but not thinking ability cultivation. This paper constructs a four-in-one teaching model of "learn-teach-show-research" for the cultivation of computational thinking ability. Through independent learning, guided teaching, accurate gap checking and challenging research, students are guided to internalize their knowledge into experience, ability and improve their data analysis and programming skills. Practical results show that by training students to think computationally, they can effectively develop their ability to solve complex problems in an integrated manner and their ability to think at a higher level.

1. Introduction

The core of talent cultivation is ability cultivation, that is, the cultivated college students need to master not only superb expertise, but also have the core ability to solve complex problems. The Ministry of Education (MOE) pointed out in the *Implementation Opinions on the Construction of First-class Undergraduate Curriculum* that the quality of curriculum directly determines the quality of talent cultivation, and higher education institutions should establish a new concept of curriculum construction and promote curriculum reform and innovation. Yan Wu, director of the Department of Higher Education of the Ministry of Education, also pointed out that curriculum is the core element of talent cultivation and the "last mile" that reflects the concept of "student development as the center" [1].

Computational thinking is a series of thinking activities that use the basic concepts of computer science to think, analyze and solve problems [2,3]. In March 2006, Prof. Yizhen Zhou published his article *Computational Thinking* in the journal of ACM, pointing out that computational thinking should not only belong to computer scientists, but should be a basic survival skill for everyone. Professor Zhou believes that computational thinking is human thinking, not machine thinking. It is a conceptual thinking, not a procedural thinking.

Programming courses are mandatory basic courses for undergraduates. They are the core courses for cultivating students' innovative ability and computational thinking. However, most of the traditional programming courses stay at the level of knowledge and concept mastery, and are insufficient in strengthening the cultivation of students' computational thinking. How to integrate the cultivation of computational thinking into the teaching of programming courses is a key problem that needs to be solved in the teaching of programming courses at present.

This paper takes the C language programming course as an example to develop students' comprehensive ability of writing programs to solve problems, while developing advanced thinking as the guide to carry out research and practice of teaching methods to accumulate experience and provide reference for the construction of programming courses.

2. Analysis of the Current Teaching Situation of C Language Programming Course

As a university-wide public basic computer course, the C programming course is the first, highly practical programming course that non-computer science students are exposed to. The course mainly cultivates students' computational thinking ability, data analysis ability, problem solving

  DOI: 10.25236/ieesasm.2022.025

ability and basic programming ability, which not only lays a good foundation for students to study in subsequent courses, but also lays a solid foundation for cultivating cross-disciplinary talents [4]. However, the teaching status and results of the C programming course show that the students' learning effect is not optimistic. There are mainly problems in the following aspects:

2.1 Not Enough Credit Hours

According to the requirements of the *National Standards for Teaching Quality of Undergraduate Classes in General Higher Education*, each major has a total credit limit, and the C language programming course, as a basic course, is generally between 2.5 and 3.5 credits, and most of the credit hours are around 40-56. However, C language programming is a very practical course. Students have to master the syntax rules and write programs within the limited class time, which seems difficult for students who have no programming foundation. Many students are weak in problem analysis and problem solving, and can only solve some simple problems. When they encounter complex problems, they do not know how to solve them by analogy, and cannot construct knowledge based on their original knowledge and theories, so that it is difficult to propose effective solutions. Secondly, it is difficult to make students understand and get used to the thinking method of programming by only relying on in-class computer experiments. Finally, it is difficult to synchronize experimental teaching with theoretical teaching due to the influence of computer room capacity and time, which makes it difficult for students to internalize what they have learned in time and improve the teaching effect.

2.2 Insufficient Attention to the Development of Computational Thinking Skills

The traditional teaching mode mainly focuses on the quantity of knowledge students can master, so some teachers pay too much attention to the details of programming language such as syntax rules and data types in the teaching process. Students only learn knowledge passively, just knowing it and remembering it without digesting, transferring and creating it, thus the knowledge only stays on the surface and is not understood in a deeper way. Although teachers teach the process and ideas of algorithms and programming to solve problems, they do not pay enough attention to the cultivation of computational thinking, and students fail to master the method of “think like a computer to solve problems”. Students have insufficient ability to analyze information and process problems using computer-written programs, unclear ideas and directions for solving problems, and difficulty in providing their own opinions and insights.

3. Teaching Model of “Learn-Teach-Show-Research” for the Cultivation of Computational Thinking Ability

From the cognitive point of view, thinking style is the sum of human cognitive stereotypes and cognitive modes of operation; from the individual point of view, thinking style is the comprehensive performance of individual thinking depth, type, and ideas; from the student learning point of view, thinking style reflects the student's perspective of understanding things, and also determines the student's thinking and direction of problem solving, which is a key factor in the mastery of knowledge and development of ability [5]. Therefore, guiding and enlightening students to learn to think correctly is the basis for developing their comprehensive skills in solving complex problems. The basic scientific thinking of human beings for understanding the world and transforming it is mainly empirical, logical and computational thinking [6]. Among them, computational thinking was proposed by Professor Seymour Papert of MIT in 1996. Professor Yizhen Zhou of Carnegie Mellon University, USA, pointed out that computational thinking should not belong to computer scientists only, but should be a basic survival skill for everyone. Computational thinking refers to the application of concepts underlying computer science to solve problems, design systems, and understand human behavior, and is characterized by abstraction and automation.

In view of the current problems of the C language programming course in the teaching process, which emphasizes knowledge lectures but not the cultivation of thinking ability and programming ability, the teaching mode of teacher lectures and student practice is adopted [7], and a four-in-one

teaching mode of “learn-teach-show-research” is built for the cultivation of computational thinking ability (as shown in Figure 1). First, guiding students to complete the basic theory of C language programming in a MOOC to realize the complementarity of physical and online classrooms. Second, the teaching concept of ability first and knowledge second in classroom teaching is adopted, and guidance on key and difficult knowledge is provided, from which students gradually establish the concept of “programming”, form computational thinking and improve thinking ability. Third, through the practical aspects of programming, students are guided to start from “design”, establish the basic concepts of spatial and temporal complexity, understand and master the knowledge and methods of program design, implementation, testing and analysis, etc. It can promote students' association and innovation in the level of thinking. Finally, challenging learning tasks are designed to guide students to delve deeper into research and develop their comprehensive skills in solving complex problems and higher-order complexity thinking skills.

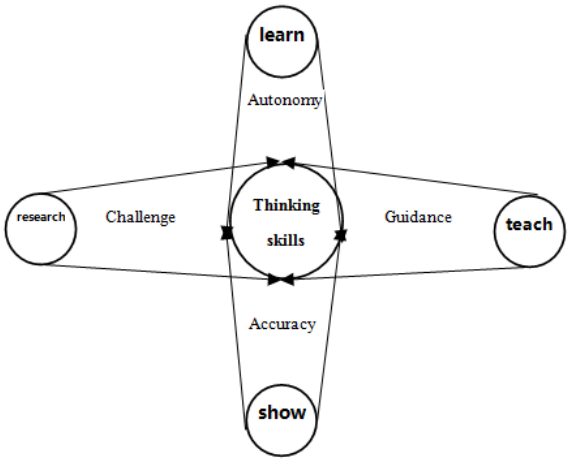

Fig.1 Learn-Teach-Show-Research Four-in-One Teaching Model

4. Teaching Practice of C Language Programming Course for the Cultivation of Computational Thinking Ability

Now taking “Designing and implementing a library information management system” of C programming course as an example to explore the four-in-one teaching model of “learn-teach-show-research” to cultivate students' computational thinking skills. The teaching case of “Library Information Management System” includes several sub-tasks: Overall system framework design (modular design), menu design for the whole system (selection structure), book price statistics (loop structure), book price sorting (array), definition and processing of book information (structure), finding of book information (loop structure) and storage of book information (file).

First of all, the teacher assigns the pre-course learning tasks in advance, and the students watch the teaching videos of “C language programming” course on China University MOOC (MOC) or Wisdom Tree for independent learning to understand and master the basic C language syntax rules, three basic structures, functions and other basic concepts and design ideas, and record the difficult and doubtful problems in the learning process. Through independent learning before class, students can strengthen their perceptual understanding of programming, and develop their active learning awareness and independent learning ability. At the same time, when teaching data types and other knowledge, teachers clarify that “data types are abstract descriptions of real-world characteristics, and there are different data types because things have different characteristics”, and consciously guide and cultivate students' ability of “abstraction” in computational thinking.

Secondly, diverse teaching methods are used in the classroom for guided teaching. First, introducing the library management system is a complex problem. If the functions of entering, deleting, finding, borrowing, returning and saving books are written in a main function, it is not only poorly readable, but also easy to make mistakes. A complex problem is usually composed of several simple problems, in order to solve the complex problem, it can be broken down into several

smaller, simple parts, that is, “modular division”, that is also, the “decomposition” ability in computational thinking. Secondly, starting from “book search”, by explaining the process of book search, students will understand that entering and exporting book information, deleting books, borrowing books, sorting book prices and other operations are all independent modules in nature. Each module completes a certain problem solving. In the program design, each module corresponds to a function, and follows the design concept of “top-down, step-by-step solution”. Next, students are guided to be able to transform the problem into an intra-computer representation by explaining the design concepts of choice structures and loop structures. Then, using arrays, structures, and files to organize data, teachers explain sorting algorithms and lookup algorithms to develop students' algorithmic thinking. Finally, the process of calling each function by the main function is explained. By eliciting problems from instantiation to how to implement them, it forms an inspiring teaching and guides students to understand and experience how to solve complex problems by computer, and realizes from knowledge transfer to thinking cultivation.

Thirdly, the interactive learning session of question-answer is designed to improve the accuracy of learning by “showing the results” to check the gaps. For the tasks assigned by the teacher, each group will send a representative to show their learning results, focusing on their design ideas and algorithm principles, so as to show the learning effect and depth of the group. The teacher and other students then ask questions. Through ask and answer, strengthening teacher-student interaction, eliminating students' doubts in the learning process, and strengthening students' understanding and mastery of easy points. Through this link, on the one hand, students can clearly recognize their knowledge gaps and learn to construct knowledge with their thinking; on the other hand, students are guided to actively think, discuss, communicate and study, from knowing knowledge to digesting knowledge, leading them to think deeply and deepen their understanding and application of knowledge.

Finally, the function of the task is extended to increase the challenge of the course. Levels of different difficulty levels are selected according to the characteristics of the teaching subjects to meet the diverse and hierarchical teaching needs of different classes [8]. For example, it is required to implement a library information management system using structure pointers. On the one hand, the introduction of pointers increases the difficulty of the task, and students must think deeply and study deeply. In the process of completing the task, students are guided to repeat trial and error, iteration, single-step debugging and other methods to solve problems, improve their program debugging skills, so that students can have a deeper understanding of array, structure, pointer, function, file and other knowledge, as well as modular design ideas. Thus, a three-dimensional and open overall knowledge structure and progressive ability structure are formed by points (knowledge points), lines (basic tasks) and surfaces (challenging tasks) [9]. On the other hand, students need to study harder to complete the task, which can stimulate students' interest in learning and research, deepen their understanding of the depth and breadth of programming principles, and develop their thinking skills and comprehensive ability to solve complex problems.

5. Summary

In the course teaching, teachers are not the movers of objective knowledge. They can not only start from a single perspective of knowledge coverage, but from the perspective of abstraction, decomposition, simulation, automation and other elements of comprehensive coverage of computational thinking, to guide students to gradually form a computer discipline perspective of thinking about problems, analyzing problems and solving them, to achieve the organic integration of knowledge, ability and quality. In this paper, the four-in-one teaching model of “learn-teach-show-research” is designed to cultivate students' computational thinking ability. On the basis of forming computational thinking, the thinking is expanded and deepened into the process of knowledge discovery and rediscovery, strengthening students' programming ability and ability to solve complex problems, and laying a solid foundation for subsequent course learning.

Acknowledgment

Project Funding: This paper is supported by the project of the Educational Teaching Reform Research Project of Shaanxi Provincial Department of Education.

Project Number: 21BY192.

Project Name: Research and Practice of a New Paradigm for Teaching Programming Classes Oriented to the Cultivation of Computational Thinking.

References

[1] Yan Wu. Building China's “ golden lesson” [J]. China University Teaching,12, pp.04-09, 2018.

[2] Jie Lyu, Ying Li, Jing Du. Practice and exploration of “university computer foundation” course reform based on computational thinking [J]. Computer Engineering and Science, 41 (S1), pp.01-05, 2019.

[3] Hailiang Deng. Exploring the reform of basic university computer courses oriented to computational thinking [J]. China Journal of Multimedia & Network Teaching, 05, pp.152-153, 2019.

[4] Wei Li, Xiaoling Shen, Lei Wang, et.al. Exploration and practice of agile teaching [J]. Computer Education, 09, pp.129-133, 2019.

[5] Wensen Yu. From effective teaching to excellence in teaching [M]. Shanghai: East China Normal University Press, 2015, pp.18-181.

[6] Lian Li. Computational thinking - concepts and challenges [J]. China University Teaching, 01, pp.07-12, 2012.

[7] Jingwen Zhou, Haifang Zhou, Di Wu, et.al. Construction of practical training resources for university computer foundation courses based on Python problem solving [J]. Computer Education, 04, pp.174-177, 2022.

[8] Guohe Li, Weijiang Wu, Jianhui Zhao, et.al. Research and practice of integrated programming course design for new engineering disciplines [J]. Computer Education, 09, pp.119-122, 2021.

[9] Wei Li, Xinhong Hei, Lei Wang. Exploration and practice on monitoring and evaluation mechanism of learning outcomes based on learning outcomes [J]. Research in Higher Education of Engineering, 02, pp.179-186, 2020.