Research and Practice of Basic Programming Course

Xun Luo[1, a], Meng Lei[1, b], Juan Li[2, c]
[1]College of Information Science and Engineering, Hunan Normal University Changsha, China
[2]College of Mathematics and Statistics, Hunan Normal University, Changsha, China
[a]luoxun@tsinghua.org.cn, [b]425675729@qq.com, [c]lxrbs@126.com

2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC) | 978-1-7281-9018-1/20/$31.00 ©2021 IEEE | DOI: 10.1109/IPEC51340.2021.9421238

***Abstract*—A new researching and teaching project of basic programming was proposed and carried out for freshmen of computer science and software engineering including the refinement of training objective, the adjustment of course schedule, the argumentation of course content, the extension of practice training, the establishment of the evaluation system and the assessment index and the aggrandizement of the usage of online practice platform and online judge platform. Finally, the effectiveness of the project proposed in this paper is validated through the case study.**

Keywords—Basic programming, coding ability, programming practice case, EduCoder

I. Introduction

Professional Ability is always the most important and main target in development of computer science and software engineering training. The development of basic abilities is always emphasized by the major of computer science and other majors concerned with it [1]. The capability of solving practice problems in the field of computer science and software engineering is also emphasized [2]. Specifically, for basic programming course, the first and most important goal is to cultivate students programming ability, and more specifically, the basic ability of programming.

The first question is what the standard of programming ability is. In other words, what aspects of knowledge and skills students have can be considered to have programming skills. CCF with Huawei and other enterprises, as well as Tsinghua, Beihang, National Defense Science and Technology University, has launched the CCF CSP (computer software capability) certification standards in 2014. The CSP takes the subject to master the program design, data structure and algorithm skillfully, through a certain range of optional general programming language, in the specified time and space, skillfully and accurately complete the programming and debugging of the given problem as the authentication standard [3]. The computer programming ability examination (PAT), sponsored by Zhejiang University, aims to objectively judge the abilities of algorithm design and program design through the unified online examination and automatic evaluation method [4]. It can be seen that the above two kinds of ability authentication would like to take a certain programming language as the carrier to master the data structure and algorithm as the connotation of programming ability.

On the other hand, programming capabilities are not limited to data structures and algorithms. According to [5], the ability training goal in the basic course of programming must include the contents of code style, code file organization and management, structured and modular programming and so on. Ref. [6] has taken the application development ability of engineering project as the ultimate training goal of programming ability. In addition, whether the special requirements in the fields of multi-thread programming, visual programming, embedded programming, network programming, database programming and so on should also be included in the scope of programming ability is also a question that can be discussed.

Since the connotation of programming ability is so widely spread, how to define the basic ability of programming has become a problem. The definition of basic programming ability directly affects the depth and breadth of the basic course of programming. Obviously, the first basic programming course cannot cover all the above knowledge points, nor can it complete the training of all the above abilities.

Another problem with the fundamentals of programming course is caused by what programming language as will be used in the course. This kind of course must choose a programming language as the carrier, so it is easy to fall into the entanglement of grammatical details and ignore the cultivation of basic programming ability for students. However, the special language as a carrier will inevitably affect the teaching objectives and contents of the course.

One approach is to use multiple languages in a course, emphasizing the commonality of programming and weakening the grammar of the language itself. Stanford University, for example, teaches computer general education for non-computer majors, mainly JavaScript, HTML and CSS programming [7]. However, the basic courses of programming for computer majors generally choose only one language as the carrier. Java is selected by Stanford University and Rutgers New Jersey State University [7, 8], while C++ is selected by Texas A & M University [9], and C is selected by HUST [10]. The same is true for most such courses in other universities. With the vigorous development of artificial intelligence and deep learning, Python language has gradually occupied a place in basic course of programming [11, 12].

This paper carries on the teaching reform of the basic programming course of computer specialty by defining the scope of the basic programming ability, refining the training objectives, adjusting the theoretical class hours, increasing the practical class hours and using the auxiliary teaching platform in large quantities.

II. SOLUTION

A. Refinement of Training Objectives

As mentioned above, the goal of basic programming course is to cultivate students with basic programming ability. Therefore, clearly defining the basic ability of programming is the first priority of curriculum reform. This course selects C language as the carrier language of programming foundation. According to the characteristics of C language, the basic programming ability is divided into two parts: skilled process-oriented programming ability and basic object-oriented programming ability. According to these two goals, the specific skills required are refined as follows. Table I is the ability connotation of the process-oriented part designed by the course.

Process-oriented programming capabilities are subdivided into two blocks. In Table I, the first 7 items provide basic grammar preparation for basic programming skills and abilities and general ways to thinking in programming design. Item 8 and 9 are mainly for training module design ability, including functional modularization and data structure, with emphasis on function design and use. Item 10 is actually preparation for the follow-up course.

TABLE I. SPECIFIC CONNOTATION OF PROCESS-ORIENTED PROGRAMMING CAPABILITY

SN	Specific content
1	Data types
2	Basic input and output
3	Arithmetic expressions
4	Control flow
5	Selection statements
6	Iteration statements
7	Arrays and Strings
8	Functions
9	Structures
10	Pointers

There are also two other areas to be explained. First of all, the list items in the table are assigned different class hours, generally less at both ends, more in the middle. For example, item 1, 2 and 9 and 10 are allocated fewer hours. And item 6, item 7, item 8 allocated more class hours, is the key, difficult content.

The second thing to explain is the teaching order of functions and arrays. The classical textbooks abroad, such as Lippman C Primer (5th Edition) and Prata (6th Edition), place arrays before functions. More generally, compound types are placed before functions, while conformance types actually include arrays, pointers, structures, and classes. The domestic teaching materials, such as Zheng Li's C programming basic course and Tan Haoqiang's C programming all put the function before the array, the chapter before the function only teaches the simple type, does not teach the compound type. According to the ability training objectives after refinement, considering the continuity of the training, the array is placed before the function, but the structure is placed after the function.

The contents reflected in Table II are easy to fall into the trap of pure grammar teaching in the process of concrete teaching practice. Such words are difficult to speak and to learn. This course uses consistent cases for teaching, see the next section.

TABLE II. SPECIFIC CONNOTATION OF OBJECTIVE-ORIENTED PROGRAMMING CAPABILITY

SN	Specific content
1	Classes
2	Access Control and Encapsulation
3	Constructors and Destructors
4	Overload of Copy Constructor
5	Overload of arithmetic operators and relational operators
6	Public Inheritance
7	Virtual function and Dynamic binding
8	Usage of private inheritance
9	Function Template
10	Class Template

B. Adjustment of Course Hours

Program Design Foundation originally as one single course has had arranged in the first semester for freshman, 64 hours for theory class and 32 hours for practice class. With the adjustment of school policy, the scheduling plan is actually problematic. Freshmen enter school late and require military training, but the term ends at the same time. So the first semester of freshman is shorter than the other semester. The basis of programming is a lot of time to practice. Therefore, according to the original schedule, the total feeling time is not enough.

Through teaching reform, the original single course of basic programming is split into two courses including Programming I and Programming II. Both of these two courses have 48 theoretical hours and 32 practical hours while Programming I is corresponding to Table I and Programming II is corresponding Table II respectively which requires to complete the C part and the objective-oriented programming part of C++ in 96 theoretical hours plus 64 practical hours in fact. The theoretical time and practical time are both increased.

In fact, this number of hours is not enough for learners to be good at C++. There are two main purposes in our projects, the basic programming skills of procedure oriented programming and the basic idea and thought of object oriented programming.

C. Content Design

As mentioned earlier, the original basic programming course is split into two courses including Programming I and Programming II both of which can be divided into two part according to the ability training targets.

The teaching content of Programming I is divided into two parts, the first part is to cultivate students' basic programming ability and programming thinking, which is the foundation of the foundation. In this part, we must always adhere to the ability-oriented design of teaching content, to put it bluntly, all the content taught in the classroom is useful for programming, not for programming. Especially to cultivate the basic ability of programming grammar content, never actively involved.

Adhere to the case as a clue teaching, for each grammatical content, provide two kinds of examples, one is the classroom explanation, the other is for students to design and complete in the practice. For example, in the selection statement, the problem how find the maximal from three integers will be explained in class. Generally speaking, after a brief thought, students can usually construct the idea of Fig. 1 to complete the task. Teachers need to explain general ideas that meet the characteristics of programming thinking, as shown in Fig. 2.

After the completion of the case, the following assignments will be arranged in practice. The first is how to find the maximal from four numbers which is quite simple since the former question has been solved. The second is sorting three numbers, which requires students to think simply. This group of cases is generally relatively simple, as this is the case used at the beginning of the course. In addition to the program function design involved in this case, another important role is to make students understand the process of programming problem solving.

```
……
if(a>=b&&a>=c) cout<<a<<endl;
```

Fig. 1. The original solution

```
……
if(a>=b&&a>=c) cout<<a<<endl;
……
```

Fig. 2. The refine solution

In addition, the case design is continuous. The case of finding maximum from N numbers can be explained in the content of loop statement by using the above case of finding the maximal of three integers in the selection statement, which can also work in the case of how to find the maximum value and the position of the maximal of N numbers in the content of array. Finally, we can use the case of finding the second largest in the practice of array, so that students can think and solve it by themselves.

The second part of Programming I aims to develop the modularity of programming. More specifically, it is to develop students' ability to use and design functions. This part of teaching is also ability-oriented, case-driven, while maintaining consistency with the previous part of teaching cases.

For example, the three numbers maximum is written as a sub function for use elsewhere. This case can train students to write the simplest function in form the basic type is returned as function parameter and function. N maximum number can write a function, this case can train students to write array as a function of parameters. Three-number sorting can be written as a sub function, a case that trains students to understand and write functions that change the state of arguments.

The first part is the design of simple class, the second part is the construction of public inheritance system. Program II also insists on teaching by case. Within the class design, four classes will be designed: integer wrapper, array, array list, and linked list. In all classes, constructors, destructors, operator overloading and so on will be involved. That is to say, repeatedly construct requirements, drive students to learn and write classes, so as to master the preliminary object-oriented programming ability. One thing to explain is that in the writing of array list and linked list classes, the use of pointers is described in detail, and the contents are added here that are ignored by Programming I.

The class of array list and the class of linked list are also useful in the second part of programming II. In the second part of teaching, the curriculum will construct a public inheritance system. Specifically, an interface IList, an abstract parent class List, and two sub-classes which are class ArrayList and class LinkedList.

In general, the biggest characteristic of curriculum content design is case-driven teaching, which trains students' programming ability in practical programming. Another feature is trade-offs, such as weakening the pointer in programming I, weakening the template and the content of some operator overloading in programming II.

D. Usage of Practice Platform

Curriculum use Educoder as a teaching support platform. Compared with other platforms at home and abroad, Educoder provides teachers with greater freedom and can organize training projects according to training objectives, which is more beneficial to teaching. The course has 10 practice items for Programming I and Programming II, as shown in Table III.

TABLE III. LIST OF COURSE PRACTICE PROJECTS

Sn	Programming I	Programming II
1	Basic IO	Simple Wrap Class
2	Relational Operators and Selection Statement	Constructors and Destructors
3	Basic Iteration statement	Overload of operators
4	Iteration statement more	Classes and Functions
5	Arrays	Array class
6	Strings	ArrayList class
7	Functions	LinkedList class
8	Function Parameters	Public Inheritance
9	Functions and Structures	Virtual Functions
10	File Access	Private Inheritance

Comparing with other training platform such as Leetcode, Hackrank and so on, Educoder is more convenient and flexible for practice case design and implementation. Firstly, Educoder provides the basic online judge facility that is supported by all of OJ platforms which means the platform can check the output of users' program automatically and feed the correspondence back to users. It is a very useful function so that amount of code training platforms support it. The point that makes Educoder out of the ordinary is Educoder can check multiple format results for users such as text, images while most of other platforms can check text format only.

Secondly, Educoder is able to support multiple files compiled project and restricted code project which means users are only able to manipulate code files that they are allowed to access by the administrator which is a very useful and effective feature in function training part and other advanced topic training parts such as classes, templates.

Furthermore, Educoder allows users to access network and database and to execute multi-thread program in its server which are not permitted in most of analogous platforms especially in C/C++ compiled environments. Those who have receive good training in Educoder can take the advance courses training in Educoder more easier such as network programming, database programming and multi-thread programming.

Each project contains several (usually 3~4) topics that are consistent or relevant. Among these topics, students are required to write a complete C program to complete the task. And some questions are actually a code complement. For example, in the practice of function, the title will give the function interface and the main function. Students must write the sub function that conforms to the interface to complete the task. This mandatory requirement is good for students to

understand and learn modular design.

Programming I and Programming II examinations are also conducted on the teaching platform. The difference is that Programming I takes exams on our own OJ, while Programming II takes Educoder exams. The key point of Programming II is to examine the modularization ability with the content of class design, so the OJ cannot be completed well and must be completed on the Educoder.

III. Conclusions

The refinement project for basic programming course first clearly defines and refines the definition of the basic programming ability aimed at which the teaching and training are carried out with the case study of consistency and relevance. Practice shows that the project has an obvious effect on improving basic programming ability for freshmen who have proved to be much more skillful and competent when going to study advanced courses.

The next step is mainly focused on the selection and setting of teaching cases. On the one hand, it is necessary to make classroom teaching cases more coherent and relevant, on the other hand, it is necessary to make students' practical cases more enlightening and thinking. Finally, it is necessary to set up improved cases for students who learn well and lead them to more in-depth professional theories and knowledge.

There are also some other future works for advanced programming projects and practice cases such as network programming, database programming and multi-thread programming.

Acknowledgements

This work was supported by the Scientific Research project of Hunan Education Department under Grant No 19C1175 and Post Graduate Education Reform project in Hunan Province under Grant No 2019JGYB091.

References

[1] School of Information Scicnce and Technology of Tsinghua University, Academic Program of Undergraduate Education. Technical Report, Tsinghua University, 2016.

[2] School of Computer Scicnce and Technology of HUST, Academic Program of Undergraduate Education. Technical Report, Huangzhong University of Science and Technology, 2017.

[3] CCF, Certified Software Professional. Technical Report, Computing Accreditation for Professionals, 2014.

[4] Zhejiang Baiteng, Computer Programming Competency Examination. Technical Report, Zhejiang Baiteng Educational Technology Co. Ltd, 2016.

[5] Meiling Cai, Jin Zhang, Lixia Zhang, Teaching Reform of Basic Course of Program Design in Focused Ability Training, Computer Education, issue 8, pp. 59-63, 2019.

[6] Chunzhi Wang, Lingyu Yan, Hui Xu, Exploration of Program Design Ability Training Model, Computer Education, issue 8, pp. 59-63, 2019.

[7] Qin Xu, Xia Zhu, Caihua Zhang, Ideological and Political Exploration of "Python course, Education Teaching Forum, issue 53, pp. 298-300, 2020.

[8] Quan Liu, Hongmei Zhang, Yongmei Zhao, etc, Exploration of "Mixed" Wisdom Teaching in Online and Offline of Computer Programming Course - Thinking Based on Live Online Teaching during Outbreak, Internet of Things Technologies, vol 10, issue 12, pp.116-117, 2020.

[9] Ming Cai, Qin Yang, Chengguang Zhang, etc, Research on Practice Teaching Mode of C Language Programming for Professional Application, Journal of Higher Education, issue 34, pp.89-92, 2020.

[10] Yuqing Zhao, Gang Chen, C programming teaching research based on cognitive task analysis, Computer Education, issue 11, pp. 147-150, 2020.

[11] Zhe Chen, Chong Wang, Zhiqiu Huang, Teaching System of Programming Course Based on Dynamic Analysis, Computer Systems & Applications, vol 29, issue 10, pp. 114-119, 2020.

[12] Yi Wei, An Online Open Course on Java Programming, Computer & Network, vol 46, issue 20, pp. 38-39, 2020.