CSCanada
Advances in Natural Science
Vol. 7, No. 4, 2014, pp. 1-5
DOI:10.3968/6086

ISSN 1715-7862 [PRINT]
ISSN 1715-7870 [ONLINE]
www.cscanada.net
www.cscanada.org

Curriculum Reform of C Language Programming and Cultivation of Computational Thinking

ZHANG Yuxin[a],*; DING Yan[a]

[a]Changchun University of Science and Technology School of Computer Science And Technology, Changchun, China.
*Corresponding author.

Received 24 August 2014; accepted 15 November 2014
Published online 26 December 2014

Abstract
In the traditional teaching mode,students passively receive knowledge, in the way that hindered the development of students' thinking. It limits training comprehensive analysis capabilities, innovation capability. The computational thinking is one of the basic objectives of teaching computer. This paper describes the methods of using computational thinking to analyze and solve problems, combined with C language programming with its characteristics, explained by an example in the theory and practice of teaching. After that, reform proposals put forward.

Key words: Computational thinking; C Language programming; Algorithms

Zhang, Y. X., & Ding, Y. (2014). Curriculum Reform of C Language Programming and Cultivation of Computational Thinking. *Advances in Natural Science*, *7*(4), 1-5. Available from: http://www.cscanada.net/index.php/ans/article/view/6086
DOI: http://dx.doi.org/10.3968/6086

INTRODUCTION
For now, teaching Mode Programming in Domestic is in high-level language to start their own system for context. The teaching key focusses statement syntax and details. Due to limitations hours of teaching, the teaching involves less Analyze problems and solutions to problems. The teaching of program design ideas is simple. Students passively receive knowledge. So, the students haven't been interested in learning. Which leads to a comprehensive analysis of student ability, innovation and practical ability are lacking. Programming Course is one of the most direct and important platform for training students in thinking ability. How the programming language while teaching students to use computational thinking church to think and solve problems has become an important topic for future teaching.

1. COMPUTATIONAL THINKING THEORY
In 2006, Carnegie Mellon University, USA, Jeannette M. Wing, in "Communications of the ACM", defined a computational thinking. Professor Zhou believes: Computational thinking is the use of the basic concepts of computer science for problem solving, system design, as well as understanding of human behavior and other activities covering a range of thinking breadth of computer science (Wing, 2006). In the Professor Zhou's views, Computational thinking is the thinking of person and isn't the thinking of machines; It is Conceptual thinking, rather than procedural thinking and it is everyone's basic skills, rather than just part of a computer scientist.

Computational thinking seeks answers by heuristic reasoning. It can be in the case of uncertain planning, learning and scheduling. For example, it uses a variety of search strategies to solve practical problems. Computational thinking accelerate the calculation by the using of massive amounts of data between time and space. For example, it has been cleverly designed in memory and external memory use. In the data compression and decompression process, which can do equilibration overhead of time and space. Computational thinking is based on prevention, protection and redundant, fault tolerance, error correction mode and a thinking system to recover from the worst-case. For example, for a "deadlock", computational thinking is learning to explore how to meet each other at the time of synchronization to avoid "competitive conditions" situation.

Here, the computational thinking emphasizes computational thinking is how to solve the problem and its implementation process and machine operation. Computational thinking learns from mathematical thinking methods, then solve the problem and design system (Denning, 2003) with computer basics knowledge. Here, Computer science is not just the hardware and software and other artifacts present in our lives. People use computers to solve problems , manage daily life,as well as communicate with others.Computational thinking is the use of heuristic reasoning to seek answers (Gros, 2007).

2. COMPUTING THINKING ABILITY AND C LANGUAGE PROGRAMMING TEACHING

Now, in the Changchun University of Science and Technology, "C Language Programming" is computer basic course to non-computer professionals. So, teaching requirements of basic courses are different from computer courses. Teaching basic courses do not need to involve all aspects of computational thinking, should need the demand for professional computer application. To train students' computational thinking skills. Now，to most of non-computer professionals students, The purpose of the learning program is not designed to be a programmer, but to learn computer analysis, ideas and methods to solve problems. So that,

In the teaching of this course, focusing on language form is dangerous. Emphasising on program design is very important in teaching. Require students that who can write programs , transfer codes. More importantly, to learn how to design algorithms. Algorithm design reflects the thinking of how it is calculated.

2.1 Innovative Teaching Methods to Stimulate Students' Interest in Learning

In the first class, we can display classic cases of good works of students pasted. Which can make students be interested in our class. In the class, we may propose the classical algorithm such as "exchange problem". Combined with real life, it like to the problem that "soy sauce" change with "vinegar". Vividly represented, students will not be confused with 3 sentences. In the basic statement level, it stimulate students' interest with this "case -driven" teaching mode. Thus, starting from the application instance, introduction of knowledge starts from the problem. Driven by the knowledge mastery of grammar rules, students be able to deep understanding of computational.

In theory teaching, we should focus on the important ideas, dilute tedious grammar rules and increase the frontier of computer knowledge. In addition, also designed some interest and attractive examples to enhance students' interest. Such as, when we teach pointer concept, we may read the story that Sherlock Holmes sent Watson to jazz Weir's room for getting order to students. So that students can understand that the pointer is pointing to a variable concept addresses.

2.2 Using Classic Case and Zero for the Whole

From computational thinking, we reorganize classic cases, so that the problem to be solved elevate to the level of computational thinking. The program "guess numbers" (Bundy, 2007), as we know. The simple the game can associate dispersed grammatical structure. For instance:

(a) "If" has a single branch selection structure, with it output the result "You are right!".

(b) Using the "if" dual- branch selection structure output the result "You are right!" or "You are wrong!"

(c) Using "if" nested structure output "It is too large!" or "It is too small".

(d) Using a loop structure, we design algorithm combined with "break" and "continue" statements. Until "You are right!", and limit the numbers of guesses.

(e) Assume that after many times, we make an array and put the five numbers in it.

(f) If "You are right!" and there are 5 numbers to guess, we use the pointer concept to achieve related function.

(g) Using structure store guesses who's class No. name and nature of information if they are right.

(h) Using file save the rankings of "guess numbers".

Thus, gradually dull knowledge into an interesting algorithm design. When the students face the interest problems, they will actively think about and discuss the various methods of problem to solve. At the same time they both master the basic grammatical structures , but also learn to use a computer program design methods to solve problems.

2.3 Inspire Students to Use Computational Thinking Methods to Analyze Problems

Such as，the problem of "Monkeys eat peaches" in C Language programming.Using the characteristics of computational thinking, we can guide students use a recursive method to solve. Under the guidance of teachers, students reverse thinking and infer from the back. N-S process is shown in char-1.In the example, Students can fully understand the process of recursive algorithms and recursive regression process.

In this process, it need that the complex problem which size is n transform the simple problem.In the problem, the count of peaches in n-th day is twice that the count of peaches in n+1-th day plus 1. At the same time, In recursive process we must consider its termination.

Until the day is ten, the number of peaches is one. In return process we obtain the simplest solution, after which return layer by layer. Gradually gets slightly more

complex solution of the problem. Because the number of peaches in 10-th day is 1, that is, the number of peaches today is twice that the number of peaches in the next day plus 1. So, *t*1=(*t*2+1)*2.

In the example, teachers guide students to use the recursive method to solve the problem by reverse thinking. Students not only master the recursive method, but also exercise of thinking. Analogous problems are the Hanoi owers problem, factorial problem and so on. Students can think of using a recursive method to solve.

Table
Monkeys Eat Peaches

Day=9,*t*1,*t*2=1	①
Day>0	②
*t*1=(*t*2+1)*2	③
*t*2=*t*1	④
Day--	⑤
Output *t*1	⑥

3. CULTIVATION OF COMPUTATIONAL THINKING IN EXPERIMENT COURSES

C Language Programming emphasis on practice and when assigning experimental tasks past, teachers unify schedule tasks to everyone. For example, do the exercises on the experimental materials step by step. In such a manner, the one hand, students who have poor ability cannot always complete the tasks and lose enthusiasm for learning and confidence. On the other hand, it is difficult that the students who have good basic improvement.

3.1 Combine Focus Experiments and Independent Experiments

When we the experiments content, in addition to requiring students to complete basic general topics, we can design some comprehensive problems for the students who have good basic. Then appropriate guidance to them.

In the experimental teaching, we can combine focus experiments and independent experiments. Uniform experiment is experimental teaching and require students to complete provisions subjects. The independent experiments refer that students are divided into groups ,in group students discuss and design of innovative experimental problems.In order to develop students ability which analyze and solve problems, first we encourage students to experiment independently to solve problems. Secondly, through communication with students and teachers, or getting relevant information by www to resolve them.

For example, in the chapter of cycle, the focus experiment problem is *s*=1+2+3+……+100. Before doing it, we can guide students to analyze *s*＝1＋2＋3. Then, it becomes more easy to solve *s*=1+2+3+……+100. After that, we can lead students to analyze s=100! Further more, the question of *s*=1! +2! +……+100！ Can be solved. Require students to discuss the design algorithm and then complete the coding and debug.

The following example reflects the progressive thinking training process after students learn cycle, array and function:

(a) Output Fibonacci number sequence using cycle;
(b) Output Fibonacci number sequence using array;
(c) Output Fibonacci number sequence using recursive function.

After the students achieve the title, teachers should summarize the thinking behind the corresponding code issues. Which can effectively improve the level of student thinking.

3.2 Experimental Problems Require Diversity Algorithm

In the designing programs , teachers should respect of each student because everyone has the premise of individual differences, for different students we should encourage to recognize the problem from a different angle, then to describe the algorithms in different ways. Using different methods to solve the problems, teachers should give proper and scientific evaluation (Gros, 2007). Which can bring up students' Diversity and innovation of computational thinking

In addition to the normal experimental teaching, we can complete practice by recognizing internships, training courses. Recognizing internships emphasis on the understanding of the course content and training courses emphasis on comprehensive training of software design. Organic combination of the two can promote students logical abstraction ability and innovation ability.

Students through experimental training can understand the essence of computational thinking. The purpose of the experimental class is to train students: (a) make the problem abstract mathematical model, (b) design algorithm, (c) write code, (d) debug, (e) operate and analyze experimental results

3.3 Design Experiment Module Combined With Different Professional Knowledge

The interest in learning computer is the best teacher. Building experimental teaching module combining for the professional, combining teaching educational psychology, which can enhance the students' initiative and extend students' thinking. The following is a comprehensive experiment designed for telecom majors:

The title is: Change the analog signal into a digital signal, then put it into memory and further programming process. In practical applications, signal acquisition circuit is completed through the acquisition and is sent to the AD converter card. AD converter converts it into a digital signal. Follow-up, we can program to the

data collected. So which realize the signal analysis and research. For example: Design a program is that carry out the data acquisition and waveform display to the 500 numbers of the AD converter received. Hypothesis, the collected data are consistent with positive selection curve under sinusoidal law. Though, the actual data collection and conversion needs to design the hardware circuit and complete data input and output via the corresponding interface. When students study the c programming, they didn't study the SCM. But here, we can do simulation and simplify to practical issues. Only requires design one function, whose return value is on behalf of the data collected every time.

4. THE BEST TIME OF COMBING COMPUTATIONAL THINKING IS THE MISTAKES WHEN STUDENTS ARE PROGRAMMING IN EXPERIMENT COURSES

When students are programming, mistakes often indicate confusion thinking. Many students make mistakes in the beginner stage.

Example:

main()

{float *a*,*b*,*c*; *c*=*a***b*; scan*f* ("%*f* %*f*", & *a*, & *b*); print*f* ("c=%*f*", c);}

On the surface, we know that the mistake is the sentence "*c*=*a***b*;". But in fact, the beginners understand the mathematical thinking and it don't reflect computational thinking. The teachers should provide enough thinking space, after that finds the problems and resolves. Our approach is: first, teachers find errors. Then according to which we propose the different mathematical thinking and calculating the difference. So we indicate the focus of computational thinking: the basic idea of how to run the program by the hardware in Von Neumann computing systems. In fact, because the program is executed from the top to the bottom by one statement. First input the numbers, then use the variables, what is consistent with the calculation of thinking in C language programming. The teachers consciously use the mistake and find the problem of thinking the way. In this way, to beginners it can make smoothly transition to computational thinking. Thus, which can enhance their ability to use computers to solve problems.

Example:

main()

{ int *a*; *a*=32767; printf ("%*d*", *a*+1);}

It will make the mistake of overflow. On the surface, it is the problem of cross-border, but the teacher just reminds students note that the range of the integer data. What becomes difficult to improve students' computational thinking. We should inspire students that the thinking of "0" and "1" in computer systems. Which include symbolic semantics, addition automation, addition of multiplication and so on. If teachers enable students to understand the reason of the mistake, their thinking will greatly expand. Which will not only cultivate students' scientific literacy computational thinking, but also improve students' interest.

5. CHANGE IN EXPERIMENTAL EXAMINATION FORM

The examination form has guiding role to students' ideas. Assessment program should reflect the importance of thinking process. For each experimental subject. If we assess students only focusing on the correctness of the final codes, students will ignore whether the idea is reasonable or not. They maybe pay much time for writing and modifying the code. For designing experiments this approach is not appropriate. From the point of view of our culture of computational thinking, to avoid the final word form according to "the result", we propose a flexible assessment methods. For example, to comprehensive experimental problems: Process design that is 40 points, coding is 30 points and system debugging is 30 points.

In specific operations, scores of all aspects can be adjusted. However, the general principle should pay attention to guiding students to pay attention to the thinking process. Setting reasonable assessment will help examine students' thinking ability. In order to promote the model of the learning of the practice of thinking, which is conducive. Imperceptibly, which can change students' concept that the code is more important than thinking.

CONCLUSION

If we combine C Language Programming and computational thinking, on the one hand , teachers from the perspective of computational thinking, redesign and organize theory teaching program to design courses. Which can improve the quality of teaching , and achieve good teaching results. On the other hand, it can also improve students' computational thinking ability. So that students are better able to apply computational thinking to solve problems. At the same time of explaining knowledge, we should teach from the "knowledge and skills" up to "thinking". The correct thoughts of calculation thinking would guide and promote long-term practice. Not only give students of fishes, but also teach them how to fish. We guide students to program using the knowledge learned for problem solving and analysis. Programming can form programming skills, eventually forming computational thinking. Obviously, the computational thinking in turn, promote the practice

of computational thinking ability. By learning algorithms and programming students can experience the process of problem for solving and design specifications and process requirements, the characteristics of the human mind coexist with computers. However, the algorithm is based on the algorithm of thinking as a starting point. Compared to calculate the theory of computational thinking , there are more limitations.

Recently, we try to use the above method of teaching the C language programming. The result is that there are good changes about student programming practice, interest in learning, learning effect and so on. Which significantly improve students' thinking ability. Computational thinking has a strong ability to innovate , developing the highest goal is to calculate the thinking ability of innovation. How to apply computational thinking in the innovation. Which is still very difficult.

Exploring new ideas and methods of teaching is endless. Thus, in the future also continue to explore and practice in teaching practice to improve teach quality to a new level.

REFERENCES

Bundy, A.(2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing, Noted Reviews, 1*(2), 67-69.

Denning, P. J. (2003). Great Principles of Computing. *Communications of the ACM, 46*(11), 15-20.

Gros, B. (2007). Digital games in education: The Design of game—based learning. *Journal of Research on Technology in Edu cation, 40*(1),23-38.

Jeannette, M. W. (2006).Computational thinking. *Communications of the ACM, 49*(3), 33-35.