2017 HAWAII UNIVERSITY INTERNATIONAL CONFERENCES
SCIENCE, TECHNOLOGY & ENGINEERING, ARTS, MATHEMATICS & EDUCATION JUNE 8 - 10, 2017
HAWAII PRINCE HOTEL WAIKIKI, HONOLULU, HAWAII

COMPUTATIONAL THINKING COURSE FOR NON-COMPUTER SCIENCE MAJORS

NAH, JEONG EUN
UNIVERSITY COLLEGE
YONSEI UNIVERSITY
SOUTH KOREA

Prof. JeongEun Nah
Yonsei University
South Korea.

Computational Thinking Course for Non-computer Science Majors

Synopsis:

Computational Thinking focuses on introducing basic knowledge of computing along with logical reasoning and problem solving skills. The curriculum consists of computational thinking and programming language with algorithmic expressions. During lab session, students practice to solve the given problems. The programming phase starts from a very easy step and progressively moves to higher levels.

Computational Thinking Course for Non-computer Science Majors

JeongEun Nah
Yonsei University

I. Background

With the development of Internets, Communications, and Technology, computer is not the option anymore but mandatory. People should be able to deal with the computers, smartphones, and the internet. We have to learn what computers are, how they work, and what they can do. Computational thinking is a thinking process inspired by an understanding of computers to solve the problems, to increase efficiency, and to communicate better with people. Computational thinking can help people to have more successful experiences in day-to-day activities.

Yonsei University has been concerned about how to provide computational thinking and decided to launch the beginning course for all students, regardless their majors. From the spring semester of 2017, we opened up a new SW field of general education and more emphasized on teaching computational thinking courses. In this paper, we introduce educational goals, course contents, and management in computational thinking and software programming.

II. The Need for Computational Thinking Course

General education courses in computer science have been traditionally rooted in some combination of topics: computer hardware, computer software programming, and computer application skills. Most of the curriculums start with learning computer programming language. Computational thinking includes a broad range of thinking tools and methods from computer science. It empowers students in all disciplines to more effectively solve problems, design systems, and understand human behavior. The goal of computational thinking is not only teaching computer programming language and skills, but also letting the students know what a computer is, how it works, and where it can be applied. Through this course, students should recognize the role of IT and sprout new ideas blending with their majors. Efforts are also needed to transform students' hard learning experiences into fun learning experiences. It is important for students to have positive self-feedback from the learning process. We expect that knowledge learned will be utilized in their majors or other fields. In this process, students will learn IT knowledge so that they raise their computer ability on their majors.

Before opening the course, we conducted a survey to prepare students for CT & SW education. The overall response rate of students was 31.3%, and 65.3% of the respondents answered that they would like to take a SW introductory course (for non-computer science major students). The 20.4% of the respondents answered that they feel strong the need of SW introductory course, and 44.9% of the respondents answered that they feel the need of it. This

is a much more positive result than we had expected before the survey. The respondents who answered that they would strongly take the course were similarity distributed from freshmen to seniors, respectively. On the other hand, respondents who said they would never take classes are 3.2%, 2.9%, 3.7% and 4.4%, respectively. Since CT courses are taken by various types of students, regardless their majors and years, professors need more attentions to their students than other classes so that any students can catch up to the end.

Before the survey was conducted, we assume that students in the humanities and social sciences would not be interested and they would not take classes even if the classes are opened. However students' expectation and demand were higher than expected. The results of the survey showed that students in the humanities and social sciences had a greater burden of completing computer related subjects. The respondents requested several conditions; diverse teaching level, small size classes, and evaluation on P/F, and taking into account the level of very beginners. As a result of the survey, the most students have the high expectation to learn the computer, and they realized the importance of computers and technologies.

III. The Design of Computational Thinking Course

The quest for the possibility of a 'thinking machine' has already been very active since the 1930s. The first scientist who presented the theoretical model of a computer, Alan Turing, constructed a computing machine that behaved like a human brain and organized it into an Automation theory. Although the history of the computer is not so long, computer has affected our lives very much. There are still few places around us that have not been influenced by computers. With IT technologies and solutions, students realize that there are many possibilities to apply, or even create them. The contents of computational thinking course include the basic principles of computer science: how information and data are processed; how computer logic is processed and operated; and how we solve the problem using computers.

The curriculum designed to focus on computational thinking laboratory practice using Python programming. Students will be able to expand on learning the application and understanding software development while learning the programming language. The curriculum includes the introducing the programming language and algorithmic expressions. The course consisted of two hours of lectures and two hours of lab practices. The two-hour lecture introduces the principles of computer science and the basic expression of programming languages. In lab session, students practice how to write programming expression to solve the given problem. The programming phase started from a very easy step and progressively moved to a higher level.

The class consists of three learning stages; understanding of main topics of computer science, expressing in computer programming language (Python), and solving the problem by writing a program(programming method). We received feedback from students that the content was difficult, but nevertheless, they followed the progress well. Before learning programming,

students learn SW introductory course first that introduces the basic concepts of computer science; logic, data structure, algorithm, modeling techniques, and parallel processing, etc. in the curriculum. The next step is learning programming languages and expressing the algorithms. Programming should be linked to the next step in lab practices. Students who have a little experience with computer languages can easily understand the programming parts and follow the exercises easily. Although the learning background is different, a well-motivated student will overcome and learn well even if it is difficult.

IV. Conclusions and Further Studies

Computational thinking course (including introductory computer science) focuses both on problem solving perspective and on understanding the principles of computer science. The course should engage students in computational problem solving and lead students to discover the power of computation in the design of their solutions. Through the learning process, students experience step by step the seeking the solutions come to real life.

We develop the course for non-computer science majors by combining computational thinking and software programming. This course is the first stage to introducing computational thinking. We are planning a survey at the end of the semester to analyze the students' satisfaction and educational effectiveness. Based on the results of the survey, we are going to adjust the level of the contents and programming practices, and we are making sure that students have a pleasant learning experience through this learning process. The next goal is that we encourage students to take advanced course so that more IT experience can be used to solve the real life problems.■