Reformation of Teaching "Introduction to Computer Science" Course at Non-English-Speaking Country

Yuxia Sun and Qingxuan Kuang

***Abstract*—In recent years, to cultivate international computer professionals and improve the education quality for foreign students, some Chinese universities have established computer science majors where all the courses are taught in English. As the introductory course to computer science majors, "introduction to computer science" plays an important role in helping students lay a professional foundation and cultivate their interest in the major. As one of the pioneering universities in China to set up computer science majors taught in English, Jinan University has ten years' teaching experience in delivering such a course. The reform and practice of teaching "introduction to computer science" in English at Jinan University is expounded in this paper from the following three aspects: (1) Developing a teaching-content system that conforms to professional training direction, adapts to the characteristics of freshmen and the teaching characteristics of a course delivered in English; (2) Comprehensively using various advanced teaching methods; (3) Creating an evaluation system with multiple assessment modes that focuses on examining students' comprehension and practical skills. The reforms and practices at Jinan University provide a reference for other universities, especially those in non-English-speaking countries.**

***Index Terms*—Introduction to computer science, taught in English, non-English-speaking country, teaching content, teaching method, assessment mode.**

I. Introduction

"Introduction to Computer Science" is an important introductory course in computer science. It aims to guide students into the world of computer science, help them learn the knowledge system, understand the direction of professional development, lay a professional foundation and stimulate their interest in this profession. But at present, computer science full-English teaching in non-English speaking countries (such as China) is still in its infancy. Accordingly, non-English-speaking countries still lack the teaching experience of the "Introduction to Computer Science" course taught in English. With the increasing demand for teaching full-English "Introduction to Computer Science" in non-English speaking countries, it is increasingly important to conduct comprehensive research and practice on the teaching of this course.

At present, the content system of the "Introduction to Computer Science" course is controversial. For the theoretical content, the course has the following three options:

1) The content is similiar to that of the "University Computer Foundation" course for non-computer-science majors;

2) The content is the compression of the four-year core courses for computer science majors;

3) The content consists of a highly abstract mathematical model used to describe the nature of computing [1].

The first option is too shallow to support the professional foundation of computer science. The second is so difficult and fragmented that students tend to learn shallowly without adequate digestion. The third option is only applicable to top universities such as Peking University, but not for students of other universities with a general mathematical foundation. Thus, an applicable theoretical content system of the course needs to be created for most computer science freshmen at average universities.

In addition, for the experimental content, the current course includes the following two options:

1) The course involves no experiments or very simple experiments (such as practicing the office software);

2) The course includes many programming experiments and the experimental hours are at least half of the course hours.

The former option, unable to distinguish the course from the ones for non-computer-science majors, makes the computer science majors hard to understand the important professional concepts based on practices. The latter option, unable to distinguish the course from programming courses, always leads to the over-simplification of the theoretical content of the course. Thus, neither of the above experimental options can play the important role of professional guidance for computer science freshmen.

As seen from the above analyses, it is necessary for us to study and reform the content system of the full-English course "Introduction to Computer Science", so to make the content system applicable to the computer science freshmen at most universities and conform to the professional goal of the major.

As a key comprehensive university in China and the paramount school of overseas Chinese students, Jinan University is one of the earliest several universities in China that set up the computer science full-English major. The first author of this paper has 10 years' experience in teaching "introduction to computer science" in English for the computer science majors at Jinan University. Based on the author's teaching experience of the course, this paper will detail the research and practice of the teaching reform of the course from three aspects: teaching content, teaching method

Manuscript received December 19, 2019; revised March 7, 2020. This work was supported in part by Guangdong Province Science and Technology Plan Project in China (#2017A040405030) and the 19th batch of teaching reform research project of Jinan University (#JG2017078).

Yuxia Sun and Qingxuan Kuang are with the Department of Computer Science, Jinan University, Guangzhou, 510632, China (e-mail: tyxsun@email.jnu.edu.cn, susan900@stu2016.jnu.edu.cn).

doi: 10.18178/ijiet.2020.10.7.1419

and assessment mode.

In the rest of the paper, first, our teaching content will be expounded, involving both theoretical and experimental parts that are carefully selected and elaborately arranged. Next, the following diverse teaching methods applied in our course will be detailed: flipped classroom learning method [2], MOOC-assisted method [3], Web-based active learning method [4] and cooperative learning method [5]. Finally, our assessment modes to examine students' learning ability and effect will be explained from multiple aspects that focus on students' comprehension and practical skills.

II. Teaching Content Reform

A. Reform Principle

For a guiding course of computer science majors in our university, our principle of setting its teaching content is to balance the breadth and depth of professional knowledge, covering both theory and practice. The knowledge system established in this course should cover the main aspects of computer science and technology, so to enable students to have a comprehensive understanding of the profession and gradually enrich the system through in-depth study in other courses in the future. When designing the knowledge points and depth of each part of the teaching content, it is necessary to consider the knowledge background of the fledglings and the adaptability of freshmen to teaching in English. The introduction of each part of the content should have a proper depth, focus on the explanation of the basic content, and pave the way for further in-depth study. For the basic content, relevant history and latest developments should be introduced. For the important content based on practice, the corresponding experimental teaching content should be designed.

In order to balance the breadth and depth of the theoretical knowledge points, we utilize two types of teaching materials: one textbook focusing on the breadth of knowledge, and the other one focusing on the depth. By combining the two teaching materials, our course can cover as many basic knowledge points as possible and explain the important knowledge points in depth.

The goal of teaching experimental content is to help students deeply understand and correctly use theoretical knowledge, cultivate practical skills, and stimulate their interest in practices. Due to the limited course hours for experiments, we must select the experimental contents elaborately. Our principle of selection is that the experimental content can enable students to understand important basic concepts or master basic practical skills, and the learning threshold is low.

B. Reform Practice

To create the theoretical content of the course, we use two textbooks, namely [6] and [7], to cover the breadth and depth of knowledge, respectively. The former textbook [6] covers the breadth of knowledge, but its explanation for some important knowledge points in computer science is not deep enough. The latter one [7] covers the depth of knowledge, but it involves a great number of knowledge points that are difficult to understand. For most neophytes majoring in computer science, it is too challenging for them to master all the difficult knowledge points. Following the principle depicted in the previous subsection, we combined the materials in both books when establishing the theoretical content of the course. As the content arrangement in the first textbook [6], we teach the following parts in the course: Ground work, Digital data representation, Computing system, Network, Software engineering, Database, Security issues, Image and Multimedia, and Artificial Intelligence. The above parts have covered most main subareas of computer science. In order to elaborate on the important professional knowledge, we also add the following parts from the second textbook [7]: Computing history, Binary Values and Number Systems (including base conversion), Gates and Circuits, Operating system (including File/Memory/CPU management), Network topology and devices. The added knowledge plays an important role in supporting the teaching of other parts of the course and subsequent professional courses: The history of computers and networking technologies is critical for students to understand the origin and development of computer science; Logic gates and circuits are the foundation for understanding the principles of computer composition; The fundamentals and algorithms of operating systems, such as file system, memory, and CPU, can help them better understand the relationship between software and hardware, master the concepts of algorithms.

Our experimental teaching content includes the following three types of experiments: programming experiments, database experiments, and demonstrated experiments interspersed in theoretical lectures. The first type is the most important one in our teaching.

1) Programming experiments

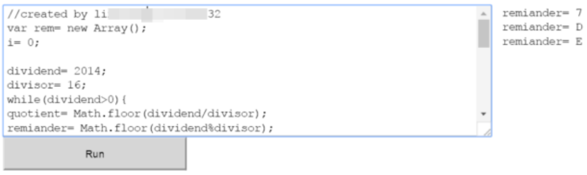
New learners in computer science are generally eager to learn programming. Since no programming courses are delivered to the computer science majors for the first semester in our university, it is more important for us to introduce programming practices in the course of “Introduction to Computer Science”, so to satisfy the students' programming curiosity and cultivate their professional interest. In addition, it is extremely difficult for students to understand the concepts and principles of the relevant theoretical parts of the course (such as software engineering, algorithms, and compilation) without any foundation for programming practice. Thus, we introduce the teaching content on programming practice to this course. In addition to the theoretical lectures in the classroom and the MOOC video learning after class (see details in the following section), we also design three experimental classes for on-machine programming.

Since this course is an introductory one, the main purpose of the programming teaching is to enable students to master basic programming ideas and debugging skills, rather than mastering a certain programming language. The programming language chosen for this course is JavaScript, rather than the language chosen for programming courses such as C or Java. The reasons include:

- JavaScript is a weakly typed language, and beginner programmers can easily understand the grammar and the debugging skills;

- JavaScript programs can be executed conveniently with only a web browser rather than with an IDE environment.

As Fig. 1 demonstrates, on a specially designed webpage we provide, the students only need to put their program source code into the code area on the webpage, press the "Run" button to perform the program, and view the results on the webpage, without having to install and learn a complex IDE environment. When we teach programming in the classroom, the focus is on helping students understand the basic JavaScript syntax, understand the relationship between programs and algorithms, and master the basic debugging skills. When designing the content of the programming experiments, we associate them to the other parts taught in the course. For example, one experiment involves the conversion between hexadecimal and decimal (as Fig. 1 shows), and another one involves image transformation.



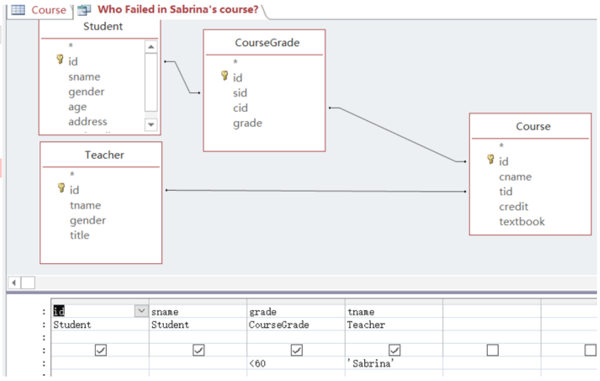
```
//created by li[redacted]32
var rem= new Array();
i= 0;

dividend= 2014;
divisor= 16;
while(dividend>0){
quotient= Math.floor(dividend/divisor);
remiander= Math.floor(dividend%divisor);
```

```
remiander= 7
remiander= D
remiander= E
```

Fig. 1. Example of programming experiment.

2) *Database experiments*

We also design several database experiments. The experiments require the students to create a simple relational database with tables and their relationship, and to query the database with SQL commands and query tables, respectively. The database experiments are conducted on MS ACCESS. Fig. 2 illustrates a student's experiment exercise that queries "Who failed in Sabrina's course?" based on a database created by the student.



id	sname	grade	tname		
Student	Student	CourseGrade	Teacher		
☑	☑	☑	☑	□	□
		<60	'Sabrina'		

Fig. 2. Example of database experiment.

3) *Demonstrated experiments*

In addition, we teach simple experimental content with class demonstration, and students can watch the related videos for review after class. One movie example is to demonstrate the usage of network diagnostic tools called "ping" and "traceroute".

III. Teaching Method Reform

A. *Reform Necessity*

The reform of teaching content described in the previous section makes the course teaching challenging due to the added parts, and the integration of theory and practice. To meet those challenges, we need adopt advanced teaching methods to improve teaching efficiency and inspire students' initiative in learning. In recent years, new teaching methods such as MOOC, flipped classrooms and collaborative learning have played an active role in teaching various computer courses. Thus, it is necessary for us to apply the new teaching methods in the course of "Introduction to computer science" taught in English.

B. *Reform Practice*

During the ten years' teaching of the course at Jinan University, we have carried out the following teaching methods gradually:

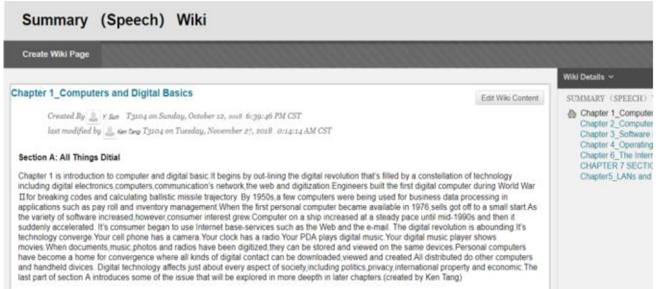
1) *Combining theory and experiment*

As described in the previous section, we select important knowledge points that are closely related to practice for experimental teaching, design a set of experimental content and requirements, and apply them in teaching. The experiments enable students to better comprehend the abstract theory, to improve practical hands-on skills, and to cultivate their interest for learning.

2) *Teaching based on course websites*

We have set up a teaching website for this course on the Blackboard system of Jinan University, which provides learning materials such as courseware, video and audio, and offers group and class discussions, exercises and questions answering. The teaching method based on the course website is convenient for students to learn online at any time, and strengthens the learning communication among students and teachers. When teaching the content of programming, we use our own network course resources (including video explanations and online interactive exercises) to guide students to preview, review and practice after class. In this way, even when we speed up explanation and teach more knowledge points in class, students can still learn solidly.

3) *Integrating wiki into flipped classroom teaching*



Chapter 1_Computers and Digital Basics

Created By Y. Sun T3104 on Sunday, October 12, 2018 6:39:46 PM CST
last modified by Ken Tang T3104 on Tuesday, November 27, 2018 0:14:14 AM CST

Section A: All Things Ditial

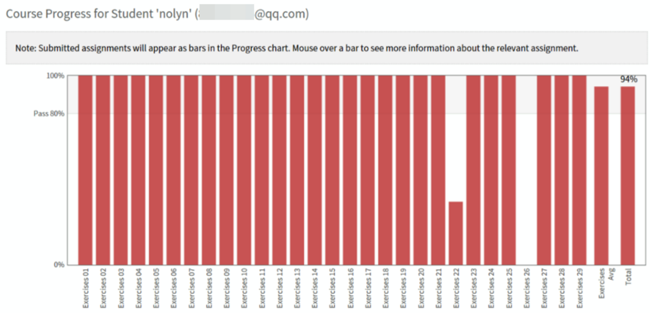
Chapter 1 is introduction to computer and digital basic.It begins by out-lining the digital revolution that's filled by a constellation of technology including digital electronics,computers,communication's network,the web and digitization.Engineers built the first digital computer during World War Ⅱfor breaking codes and calculating ballistic missile trajectory. By 1950s,a few computers were being used for business data processing in applications such as pay roll and inventory management.When the first personal computer became available in 1976,sells got off to a small start.As the variety of software increased,however,consumer interest grew.Computer on a ship increased at a steady pace until mid-1990s and then it suddenly accelerated. It's consumer began to use Internet base-services such as the Web and the e-mail. The digital revolution is abounding.It's technology converge.Your cell phone has a camera.Your clock has a radio.Your PDA plays digital music.Your digital music player shows movies.When documents,music,photos and radios have been digitized,they can be stored and viewed on the same devices.Personal computers have become a home for convergence where all kinds of digital contact can be downloaded,viewed and created.All distributed do other computers and handheld divices. Digital technology affects just about every aspect of society,including politics,privacy,international property and economic.The last part of section A introduces some of the issue that will be explored in more deepth in later chapters.(created by Ken Tang)

Fig. 3. Example of Summary (Speech) Wiki.

We use the method of "Integrating Wiki into Flipped Classroom Teaching" for teaching each chapter's summary in the theoretical parts. The method is implemented as follows: The teacher assigns the task of summarizing a chapter to each study group, and establishes a group Wiki for each group on the course website, called "Summary (Speech) Wiki", as illustrated in Fig. 3. One student in a group submits his/her summary to the group Wiki, and all the other group members co-edit the summary. The final version of the summary will be completed cooperatively by the group members, and posted on the group Wiki. Before a new chapter is taught, at least one group is responsible for posting the prior chapter’s summary to the group Wiki that is accessible by the whole class, and for giving a summary speech in class. By this means, the flipped classroom of "one group gives lectures, and the other groups learn" is realized.

4) Initiative learning based on MOOC and other network resources

In addition to the self-built resources on the course website, we also make use of two interactive training resources on the external network to teach students to arrange their studies independently through the network. One external network is Stanford University's programming MOOC network [8], the other is online programming exercises in the form of games [9]. We guide students to use the teaching resources on the MOOC network, where students can arrange their own learning schedule and get feedback from the MOOC exercises. Over 95% of our students learned the MOOC lectures after class and completed the first 10 exercises that are required by us. Around half of the students finished all the exercises online including optional ones. Fig. 4 shows an average student’s exercise result on the MOOC system. When the second external network is used in online programming practice, the student's learning initiative is effectively stimulated due to the game characteristics of the exercises.



Course Progress for Student 'nolyn' ([redacted]@qq.com)

Note: Submitted assignments will appear as bars in the Progress chart. Mouse over a bar to see more information about the relevant assignment.

Fig. 4. Example of student’s initiative learning using Stanford MOOC.

5) Web-based collaborative learning

We use the Blackboard system to create a collaborative learning environment on the course website. We design collaborative learning tasks, so to realize the collaboration and interaction among students and the teacher. The detailed cooperation and interaction are depicted as follows: First, a student cooperates with other students by using group Wiki and class Wiki to complete the group tasks and class tasks assigned by teachers; Second, a student communicates with the teacher or other students by utilizing BBS discussion and question-and-answer area on the course website. In this way, the students can help each other and promote each other in their studies. Moreover, during the teaching of this course, the teacher and the students have merged a knowledge construction environment into the learning process.

IV. Assessment Mode Reform

In the teaching of this course, we have also attempted to reform the way of course assessment, focusing on examining students 'learning ability and study effect from various perspectives, and comprehensively examining students' comprehension ability and practical ability. Our course-assessment system involves the following two parts:

1) Assignment Evaluation: Assignments of this course have multiple forms with different evaluation goals, including evaluating the students' theoretical learning effect or the experimental results, assessing the students’ participation in the flipped classroom or the cooperative learning. We record and evaluate students' assignments primarily based on the course website and the MOOC exercise system.

2) Final Examination: The closed-book final exam focuses on two parts: examining students' understanding of basic concepts and principles, and using algorithms to test their problem-solving skills. For example, in the final exam of the course in 2018, the former accounted for 69% of the score, while the latter accounted for 20%.

V. Analysis of Teaching Effect

In the recent ten years, as described in the previous sections, we have been conducted a series of teaching reforms in the course of “Introduction to computer science” that is taught in English for the computer science majors at Jinan University. The students involved in the teaching reforms are more than 200. With years of continuous reform, our teaching effect of this course has been constantly improved. For example, the students' average evaluation grade for our course teaching has reached 91 points (full grade is 100 points). According to the students’ feedback in the course survey, 99% students feel that the practical parts of the course are not only interesting but also instructive to understand the theory; 94% students think that the teaching resources on the course-website and the MOOC system can help them learn by themselves.

VI. Conclusion

“Introduction to Computer Science” is an important introductory course for computer science majors. It is challenging to teach such a course in English at a non-English-speaking country, such as China. Based on ten years’ experience of teaching the course in English at Jinan University, we summarize our research and reform of the course on teaching content, teaching methods and assessment modes, respectively. In the future, we will continue to explore and solve various teaching problems of this course through practice.

Conflict of Interest

We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Author Contributions

Yuxia Sun designed and performed the teaching reform, conceived and wrote the manuscript. Qingxuan Kuang participated in the reform assessment.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their insightful comments.

REFERENCES

[1] Z. He, "Research and practice on introduction to computer curriculum content system construction," *Journal of Hunan First Normal University*, vol. 10, no. 5, pp. 63-66, 2010.

[2] X. You, Z. Fang, and X. Yao, "Teaching reform and practice in applied universities under MOOC+flip CLassroom hybrid teaching model," *Software Guide Educational Technology*, vol. 16, no. 1, pp. 7-9, 2017.

[3] Y. Nan and C. Guo, "Flipped classroom teaching design of computer curriculum under the background of MOOC," *Heilongjiang Science*, vol. 7, no. 17, pp. 120-121, 2016.

[4] X. Wang, Q. Liu, and R. Zhang, "Inspired by ISEC project: Cultivation of engineering students' active learning ability," *Journal of Chifeng University: Natural Science Edition*, vol. 33, no. 3, pp. 191-192, 2017.

[5] K. Zhu, B. Li, and L. Su, "Trigger mechanism and empirical study of collaborative learning in e-learning space," *Audiovisual Education in China*, vol. 7, p. 4, 2018.

[6] J. J. Parsons and D. Oja, *New Perspectives on Computer Concepts, Comprehensive* (China Student Edition), 15th ed. China Machine Press, 2013.

[7] N. Dale and J. Lewis, *Computer Science Illuminated*, 6th ed. Jones & Bartlett Learning Press, 2015.

[8] Engineering: CS101. Stanford Online. [EB/OL]. [Online]. Available: https://lagunita.stanford.edu/login?next=/courses/Engineering/CS101/

[9] CodeCombat: Learn how to code by playing a game. [EB/OL]. [Online]. Available: https://codecombat.com/

Copyright © 2020 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (CC BY 4.0).

Yuxia Sun received the B.S. degree from the Department of Computer Science, Huazhong University of Science and Technology, Wuhan, China, and the Ph.D. degree from the Department of Computer Science, Sun Yat-sen University, Guangzhou, China.

She is currently an associate professor with the Department of Computer Science, Jinan University, Guangzhou. She was a research associate with Hong Kong Polytechnic University, Hong Kong, and the University of Hong Kong, Hong Kong, and a research scholar with the College of Computing, Georgia Institute of Technology, Atlanta, GA, USA. She has published over 20 academic papers on international conferences or journals including IEEE Internet of Things Journal, Chinese Journal of Computers, and so on. Her current research interests include software engineering, software safety, and system safety.

Dr. Sun is a member of IEEE, ACM, and CCF. She received a Ministry of Education Nominated State Science and Technology Award in China in 2005.

Qingxuan Kuang is an undergraduate student major in computer science at Jinan University, Guangzhou, China. She is a teaching assistant and a research assistant in Jinan University, admitted into Jinan University in 2016. She is the PI of a project on Block Chain National innovation and entrepreneurship training program for college students. Besides, she has obtained two patents. Her current field of study includes software engineering.