

# A Student-centered Learning Mode for Teaching Deaf Student Programming based on Computational Thinking

Xueyan Dong<sup>1\*</sup>, Jingpeng Jia<sup>1</sup>

<sup>1</sup>Beijing Union University, Beijing, China

\*Corresponding author. Email: tjtxueyan@bnu.edu.cn

## ABSTRACT

Programming is an essential skill for students majored in computer science. However, due to the effect of sign language, deaf students have more difficulties in learning programming than average students. A primary reason is that they fail to understand programming logics and have poor English skills. Therefore, we reformed the teaching mode in the programming course by borrowing ideas from computational thinking. The new teaching plan includes concept abstraction, task decomposition, independent thinking, and practice reinforcement. The aim of the plan is to let students take the dominant role in the class so as to change passive learning into active learning. The new plan was applied in a programming course for one semester, the students' performance in the process of classes and the results from final project show that most of students are fully activated and achieved from the course.

**Keywords:** *Student-centered learning; Deaf students; Programming; Computational thinking.*

## 1. INTRODUCTION

Traditional teaching classes are teacher dominated lectures and tutorials, so students passively accept what being taught by teacher. It leads to many problems, including students lacking of interests for the content, less participation in the class, low creativity passion. These problems invoke many educators to rethink what are really effective education. Is it individual show of teacher, or the tool to get points for a degree? The answer is definitely neither. Teachers are supposed to be assistant and tutor for students. The aim of university education is to serve students so as to fulfil their ideals. Therefore, many educators proposed to take the student-centered learning into the class [1][2].

Based on such a background, many universities made a lot of efforts to explore the appropriate means and teaching mode benefit for students. To realize the student-centered learning, teachers have to make an agreement on the roles of teachers and students. In the new learning mode, students are supposed to take the dominated role because they are learners. Teachers are tutors to provide advice and instruct for them. The aim of student-centered learning is to guide students for active learning [3].

To support student-centered learning, technology is used in the class to assist the learning process in automatic way [4]. Besides, student-centered learning is regarded as a good way for cultivating students' leadership [5]. Recently, our university proposed a suggestion on implementing the student-centered learning environment and teaching mode. To realize the suggestion, we modified the teaching methods in real course.

In this study, our new teaching methods were put into practice in programming course. Specifically, we first analyzed deaf students' learning habitats and made corresponding teaching program. In our program, we are especially inspired from the concepts in computational thinking. Computational thinking is a competitive skill in the early 21st century [6]. It provides new ways to address the complex problems in the real life. Especially in terms of problem solving, Sharifah pointed out that CT can provide helpful ideas to solve a problem by breaking down it into smaller problems, which is called decomposition in computer science [7]. In addition, in computational thinking, abstraction is another important skill to learn. It requires learners to turn the solution to problem into some key terms and algorithm so as to write code for implementation. Thus, three processes are required,

which are problem formulation, solution expression and solution execution [8][9][10]. However, for our deaf students, the abstraction procedure is a great challenge, nevertheless the other two processes primarily because they tend to be influenced by their first language, which is sign language [11]. Therefore, the existing abstract terms have to be translated into sign language for them to easily understand [12].

In the traditional classes, teachers give lectures by explaining programming grammars and theory. In practical classes, lots of practices are made in a black screen programming environment. However, such a setting, students can hardly write their own codes rather than solve problems by programming. To improve the case, we redesigned teaching methods referencing ideas from computational thinking to help deaf students better learn programming. After one semesters' experiment, more deaf students attempt to adopt automatic programming skill to address real-life problems.

In this paper, we will first introduce our teaching plan in section II. Section III gives an example of how to apply the teaching plan in real classes. In the end, we discussed our plan and give the conclusion.

## 2. STUDENT-CENTERED TEACHING MODE

To achieve the student-centered principle, we modified our teaching mode in four aspects shown in Figure 1.

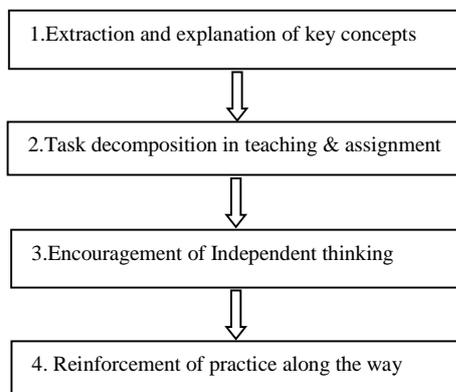


Figure 1. Process-based design of class.

### 2.1 Extraction key concepts

Rather than list out all the grammars of the programming language in the past, we extract key concepts and terms from the language and explain them as points in the lecture. Then, students are encouraged to learn others after the class. And this make students' self-learning capability to be trained.

More importantly, we state the principle less is more. It seems that less content is present in the course

than before while students can understand more and deeper. It shows that we put students' understanding in the first place.

Especially, to teach deaf students programming, it is more important to reinforce understanding of concept. Due to the differences between sign language and written language, there is always misunderstanding when introducing new concepts in each class. It is suggested by a sign language expert as she believes any subjects for deaf students has to solve the problem of understanding the written language. Otherwise, teaching programming becomes less effective.

In the real class, we put more emphasis in the first class for every week tutorial.

- First, we emphasis on key concepts and terms, this means pre-learning and pretest is quite necessary. Because deaf students may not know even quite average terms (such as abnormal) that we considered. Therefore, knowing their basic levels is quite important.
- After doing the word test before the classes, teacher need to explain it well using sign language. The most important thing is to ensure all students agree with teacher's explanations. Only in this way can students do what they are supposed to do.

### 2.2 Task decomposition

We believe learning for purpose is a very effective way, especially true for learning programming. Programming is a magic tool because it can address some specific problem, therefore, we collected a list of tasks before the class, and make them as assignments or projects.

To help deaf students gradually master the core of programming, teachers have to a lot of work in preparing teaching. We take the concept of task decomposition from computational thinking. For each assignment or project, a task is given to students.

The key to realize it contains two points:

- The theme of the task is always students' life, so they are very familiar with it, and no need to worry about unknowing about the workflow.
- To lower the difficulties, the tasks are divided into several small steps with instructions and tips. More importantly, teachers have to be very carefully to put some challenges step by step so that students can hardly feel difficulty.

### 2.3 Independent thinking

For deaf students, they tend to accept the forced-feeding method of teaching before entering university education. Therefore, they are more likely to be humble and not willing to take challenges and think

independently. As we know, independent thinking is an essential skill for students in the 21st century. To help them cultivate such a skill, we took two actions.

- Set open questions for each tutorial. We tell students that there is no right or wrong answers and they just need to try to find possible keys on their own.
- We also encourage them to share the story behind the searching work. The sharing opportunity is given to every student not just excellent one.

**2.4 Practice reinforcement**

Practice makes perfect, it is especially true for the programming course. We set more assignments the assignment than before, but this does not mean it takes students more time or effort, but more challenging and exciting. As there is no force for them to do all of them, it is student’s right to decide to take which one.

Also, the forms of assignment become various. It includes online quiz for testing the grammars on programming language, and project for examining the coding skills to solve problems, and report for summarizing what they learned and answers they gathered to the challenging question.

**3. APPLICATION**

This section will demonstrate an example of application of reformed teaching plan in a web development programming course.

**3.1 Course introduction**

According to the syllabus for 2017, the web development programming course is an optional course for junior students majored in computer science. This course aims to teach students how to establish a full stack website using JavaScript and PHP. The purpose of the course is emphasis on the understanding the programing skills and students’ abilities to application. The prerequisites include basic web development, dynamic web page development, and database technology.

In the teaching plan, we focus on process assessment and choose the hundred percentage point system. The course examinations are divided into three parts in percentages: attendance takes 10%, weekly assignments are 60%, and final assignment accounts for 30%. As you can see the weekly assignments takes the majority part in the total points. Specifically, it contains online quiz for 30%, code practice for 20%, team report for 10%.

Next, we will take one chapter in the course as an example to introduce how our teaching plan is implemented in the real course in detail.

**3.2 The implementation of teaching mode**

The lecture for each chapter is divided into before-class, the in-the-class, and after-class, see Figure 2.

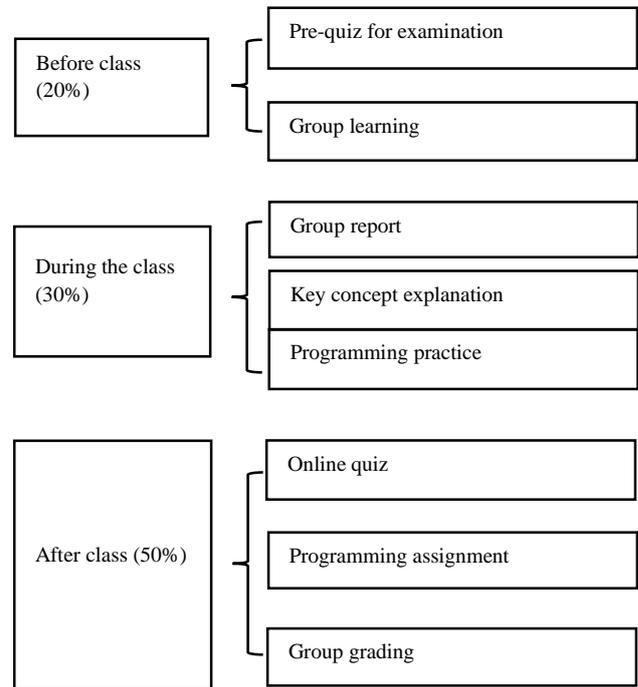


Figure 2. Process-based design of class.

Considering the teaching hours in the class is quite limited, we believe more self-taught should be given out of the class. So, the new teaching plan put more emphasis on pre-course than the old one. In addition, students themselves are encouraged to give presentation or report on their results of group study. This changes the traditional teaching way where the lecturer was the dominator in the class, and students are accepted passively. To boost their understanding, more practice is required after the class.

Next, more details will be given on each section.

**3.2.1 Before-class**

This section is very important for exercising the self-learning ability for students. In order to attract students’ attention on this part, a list of compulsory tasks are given as below.

- Pre-quiz: the purpose of the chapter quiz is to find out the possible concept gaps before the new class.
- Group learning: to make the self-learning more targeted, 3-5 questions are always given at the end of the last lecture, so that each group should attempt to find answers for all of them and prepare one random public report in the next lecture.

### 3.2.2 *In-the-class*

We made great changes in this section. In the past, the lecturers give the presentation on new content for the half of course and the remaining time are designed for practicing with guidance. We found that students can easily lose their interests, so we changed our plan in three aspects as below:

- Group report: each group is asked to report their self-learning results for other students. And when they are reporting, other students can pose questions. Notice that in this stage, the lecturer becomes the audience and assistant. This means students are dominated in the classes so as to be more participated into the class. The lecturer can also share his own experience with students but not teaching.
- Key concept explanation: after the group report, the lecturer will bring up new concepts and terms in the new chapter. And also assign the task for the practice section.
- Programming practice: students are required to use the learned skills to complete the task, and the lecturer need to help them fix the bugs.

### 3.2.3 *Assignment after class*

To give more rights to students, we provide various types of assignments which are presented in below:

- online quiz: after the class, the lecturer will release the online quiz on the app. The quiz mainly aims to examine students' understanding on new concepts.
- Programming assignment: for each chapter, a comprehensive task is given to apply the programming skills. To make it more challenging, the task can be finished either individual or group. For the group, the points will be given to everyone on average.
- Grading for group report: according to the group report in the class, every student also needs to give fair and objective scores for other groups.

### 3.2.4 *Summary and reflection*

In the past, the summary and reflection are completed at the end of the whole course, so that the feedback will be added to the teaching plan in the next semester. However, we found that it is too late to adjust the plan. Because the feedback can be quite different when various students take the course. The most effective way is to summarize the experience timely and apply it to the subsequent chapter in the current course.

Therefore, our lecturers need to modify the teaching plan at any time. But there is no worry about putting more effort for the course because the judge of most assignments can be done automatically using the

teaching assistant app, which saves a lot of time for teachers.

## 4. CONCLUSION

Student-centered is taken as an important teaching principle in Beijing union university. To realize the principle, teachers are not the dominant role but the students in the class. Students are not only accepting knowledge and selecting what they are interested. Therefore, we changed our teaching mode by reducing lecturers' presentation and increasing students' reports. Also, we take the problem-solving based program into the class, which can enhance students' subjective consciousness.

In summary, our teaching mode has three innovative changes. First is always explaining the key concept first. Which aims to address the misunderstanding due to language differences. Second is to give more opportunities to students and let them feel dominated roles in the class. Thus, they can be participating the class with more passion. Last but not least, many concepts from computational thinking are introduced into the programming course, so that reduce the learning burden and build ladders to help students gradually master the programming skills. The final project shows that more students like programming using the new teaching mode. In the future, the teaching mode will be put into other programming course for extension application.

## ACKNOWLEDGMENTS

This work is financially supported by the 2020 Premium Funding Project for Academic Human Resources Development in Beijing Union University (Project No. BPHR2020DZ05). Specially, we would like to thank Professor Lv, who is an expert in Chinese sign language research, for giving us great advice on how to help deaf students understand programming language in the design of teaching scheme.

## REFERENCES

- [1] Chen, R. J. (2010). Investigating models for preservice teachers' use of technology to support student-centered learning. *Computers & Education*, 55(1), 32-42.
- [2] EE Peters. (2010). Shifting to a student-centered science classroom: an exploration of teacher and student changes in perceptions and practices. *Journal of Science Teacher Education*, 21(3), 329-349.
- [3] Beichner, R. J. (2011). The Student-Centered Active Learning Environment for Undergraduate Programs (SCALE-UP) Project. Joint Spring 2011

Meeting of the New England Sections of the APS and the AAPT. American Physical Society.

- [4] Moeller, B., & Reitzes, T. (2011). Integrating technology with student-centered learning. education development center inc.
- [5] Haber-Curran, P., & Tillapaugh, D. W. (2015). Student-centered transformative learning in leadership education: an examination of the teaching and learning process. *Journal of Transformative Education*, 13(1), 65-84.
- [6] Council, N. (2010). Report of a workshop on the scope and nature of computational thinking. national academies press.
- [7] Barth-Cohen, Lauren, Huang, Xiaoting, Shen, & J. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & education*.
- [8] Djambong, Takam. (2016). Task-based assessment of students' computational thinking skills developed through visual programming or tangible coding environments. International Association for Development of the Information Society.
- [9] Repenning, A., Webb, D. C., Brand, C., Gluck, F., Grover, R., & Miller, S., et al. (2014). Beyond minecraft: facilitating computational thinking through modeling and programming in 3d. *IEEE Computer Graphics & Applications*, 34(3), 68-71.
- [10] Rubinstein, A., Chor, B., & Fox, J. A. (2014). Computational thinking in life science education. *PLoS Computational Biology*, 10(11), e1003897.
- [11] Israel, M., Wherfel, Q. M., Pearson, J., Shehab, S., & Tapia, T. . (2015). Empowering k-12 students with disabilities to learn computational thinking and computer programming. *Teaching Exceptional Children*, 48(1), 45-53.
- [12] Dong X.Y, Jia J.P. (2020). Teaching reform of software engineering course based on computational thinking, International conference on artificial intelligence and computer engineering (ICAICE).