Statement of Teaching

Weiwei Chen

I was inspired to be a computer scientist not only because computers are widely used in almost all the disciplines nowadays, but also because they make theoretical knowledge practical and beneficial to human lives. To well prepare the students for their future career in the area of computer science and engineering, I believe solid fundamental knowledge and higher order creative and critical thinking skills are two key factors. Therefore, there are three tenets in my teaching philosophy: 1) actively engage the students in the classroom to efficiently acquire comprehensive knowledge in the discipline, 2) bridge the gap between abstract theories and practical applications, and 3) help students develop critical and creative thinking skills to face real-life challenges.

My first goal for teaching is to actively engage the students in the classroom for efficient knowledge learning. This is based on the belief that learning is an active process which happens internally in the head of the learners, and can be remarkably stimulated and influenced by external agents such as teaching. Studying computer science and engineering needs an enormous amount of knowledge and information. Traditional education is more lecture-centered since lecturing is the most efficient way for information delivery. However, it is not a very efficient way for students to learn. I recall my own learning experience in college where education was very traditional at that time. I spent a lot of time outside the classroom by myself to process the information and figure out the connections to truly comprehend the subjects. I believe that learning can be more efficient and effective if the students themselves can actively be engaged in the classroom.

I have tried several active learning strategies in the advanced C programing course that I taught in the University of California at Irvine. My teaching duty for this class was to lead weekly laboratory sessions where the students were asked to finish a quarter-long programming homework series. Instead of lecturing the students on what and how to do the homework, I used learning activities as the major approach for teaching. Before each lab session, I designed several sets of questions regarding the corresponding concepts and the steps that may be helpful for designing the program. Then, in the lab, I used the three-step jigsaw discussion activity to facilitate active learning. First, the students were asked to work on one set of the questions alone for several minutes. Then, they shared their thoughts for the same question set in groups. In the last step, I regrouped them with the students who had worked on different question sets previously so that they can teach each other their understanding on the corresponding questions. I joined them during group discussion to clarify some confusions, and brought them back together afterwards for debriefing.

This learning activity forces the student to actively reflect what they have learned in the class. By answering the questions step by step, the students get to experience the thinking process of designing a program as well as building the connection between theoretic knowledge and programming practice. The communication skills and teamwork spirit are also honed through group discussions and peer evaluations. This teaching experiment is proved to be very successful according to the score of 3.93 (out of 4) for my teaching evaluation, and some encouraging comments from the students, such as "the way she runs the lab is very good" and "I like how you conduct group activities in class to help us gain a grasp of the assignment".

My second goal for teaching is to bridge the gap between abstract theoretical knowledge and real-life experience. Computer science theories are often rigorous and abstract to college students. Working on real-life problems helps to make the students feel what they are learning is related to their life and background. In real teaching practice, I can use case studies in my lecture or design homework assignments to solve real-world problems. For instance, in an introductory programming course, I prepared a problem for simulating the demographics of rabbits, deer, and wolves in an ecosystem in a certain number of years. The students can then practice using the loop statements in their program to solve this problem. I believe that solving real-life problems provides students emotional involvements with the course subjects. It is proved to be helpful to enhance the learning results in ["Leaning as Biological Brain Change", Robert Leamnson, 2000].

Finally, critical thinking is one of the most important skills for a good computer scientist or engineer. My philosophy is to always emphasize the thinking process and the development of problem solving skills in teaching. As I already practiced as a teaching assistant, I will tell my students on the first day of class that thinking is one of the most important skills that they will learn from the course. In the classroom, I will continue to use active learning strategies, such as think-pair-share, group discussion, and peer evaluation, to facilitate the process of learning. During office hours, I will address the student's confusions by asking them step-by-step questions, encouraging them to brainstorm the possible solutions, challenging their answers to push further thoughts, and providing frequent praise as encouragement. Moreover, most of the problems we face in life are ill structured. We need to help our students to develop the skill transferring ability under different circumstances. I will encourage and facilitate undergraduate students to participate in research projects to leverage their academic knowledge to solve real world problems.

In the past four years, I have been working as a teaching assistant for six undergraduate courses and the substitute lecturer for one graduate course. Most recently, I designed and led the day-and-a-half TA training program with eleven workshops regarding pedagogy, teaching practice, and TA knowledge, as part of my Pedagogical Fellowship. In addition, I am also serving as a resource for the teaching assistants in the School of Engineering and offering teaching consultations to help TAs promote the development of their teaching skills during the academic year. I enjoy working with my fellow TAs to foster greater teaching standards. Also, as part of the Pedagogical Fellowship, I have completed a three-quarter course series on advanced pedagogical skills and curriculum design with an emphasis on student-centered learning. In these courses, I had the chance to read a substantial amount of research work on pedagogies in higher education, and discussed with my fellow graduate students who share the same passion on teaching from different disciplines. I have developed an avid interest in innovative pedagogical practices in computer science education. I would relish the opportunity to develop my own curricula to apply different pedagogical strategies for computer science.

I believe the ultimate goal for teaching is to prepare our students to be a lifelong learner in this changing world. No matter in academia or industry, our students will have to learn independently, and come up with their own ideas to solve problems. In my opinion, the hallmark of a good teacher is one who can help students actively acquire fundamental understanding, professional skills, acute intuitions, and strong self-confidence in their respective fields. I sincerely hope my teaching philosophy indicates my desire and commitment to pursue the excellence in every endeavor, and your kind consideration to give me the opportunity to bring my passion on education to your institution.

Teaching Awards

- Pedagogical Fellowship, University of California at Irvine, 2012-2013
- Excellent Teaching Assistant Award, Shanghai Jiao Tong University, 2006

Teaching Experience

- Teaching Assistant: Computational Methods in ECE (EECS 10, 4 quarters), Advanced C programming (EECS22 and EECS 22L, 3 quarters), Digital Integrated Circuit Design, Design Automation for Integrated Circuit, Embedded System Design
- Pedagogical Fellow: Teaching Assistant Professional Development Program (TAPDP, 10 workshops)
- Substitute Lecturer: Advanced System Software (EECS 211, 2 lecture sessions), System-on-chip Software Synthesis (EECS 222C, 1 lecture session)
- Lecturer: Calculus, Linear Algebra and Theory of Probability at SAT-AP level.

Teaching Preferences

I feel comfortable to teach the following courses, and am willing and able to teach others to meet departmental needs:

C/C++ programming,
Data structures and Algorithms,
Operating Systems,
Computer Organization and Design,
Embedded Systems regarding modeling approaches, system-level description languages, validation and simulation, and system synthesis.

I intend to design the courses that emphasize both theory and hands-on practices.