Teaching Statement

Jun Li

My passion for teaching courses in computer science and engineering comes from my passion for computers. When I was a kid and touched a computer for the first time in my life, I was amazed by what a single computer can do. Since then, I have seen how much my daily life can be changed by various computing devices and many things in our life essentially depend on the power of computers today. Now I have taught different courses in computer science and engineering as a teaching assistant. When I see students sitting in my class, I see their curiosity and eagerness for discovering the magic of computers, and I strive to evoke the same excitement that I felt many years ago.

My first goal of teaching is to pique my students’ interest, just like the computer caught my attention many years ago. I believe that education should not be a daunting journey. Instead, it should be an adventure with fun and new discoveries. To achieve this goal, I strive to intrigue my students by carefully choosing the materials. I use demos and videos to give students an intuition on how a great idea works. I choose interesting materials, including stories or anecdotes of famous people in both computer science and the IT industry, to create a fun atmosphere. Further, I usually associate the knowledge with students’ own experiences by using examples from their daily life, such as applications or computing devices, to share the wisdom. As a rule, I find students very interested in such examples that directly refer to their daily life. I use all these materials to control the pace of my teaching so that students can keep their curiosity in learning without losing their interests.

When I take courses in computer science and engineering, I find it vitally important to build a solid foundation of basic concepts. Hence, I apply this principle to my students. In the fall between 2012 and 2014, I was a teaching assistant for Programming Fundamentals and Introduction to Programming in Python, two courses for the first-year students at the University of Toronto. For most of the students taking this course, this was their very first experience of learning programming. Because of this, most questions from students were asking for help on debugging. Instead of simply correcting their code, I considered such questions good opportunities to strengthen the essential knowledge and get a better understanding of the programming language. Therefore, I always reviewed the related concepts first with students and encouraged them to instantly practice such concepts by analyzing their code to find the problem. In fact, though students might spend more time at the beginning to master the essentials, they would do better in their future labs, assignments, and exams, and even in their future courses, as they had built a solid foundation in both theory and practice.

In addition to mastering the knowledge, I think that problem solving is also an important skill to become a good computer scientist or engineer, as techniques can evolve very fast and new problems emerge with no solution given in advance. Therefore, I inspire students to learn new knowledge to solve their own problems, especially when teaching senior students. When I was a teaching assistant and taught Distributed Systems, a course for students in their fourth year, students were required to implement an interactive online game and ensure the consistency among
all players in the course. Many students asked me in my office hours to discuss the designs of their projects. Very soon I found that most students were not feeling comfortable with problems having a lot of details to be specified, and the most frequent question was if their design was correct or expected regarding some details. Therefore, as there was no standard answer regarding the specific details, I asked students to specify their own scenario and discussed with them if their design met the requirements of the scenario. In most cases, such scenarios did not have a typical algorithm to solve and I encouraged students to design their own algorithm for this particular scenario. If students could not find a good algorithm working for such a scenario, I helped them to look for research papers. In this process, students learned how to specify details in a problem that was not well defined; how to find the knowledge fitting the particular requirement; and most importantly, how to solve problems in the real world, rather than well-defined problems in the class.

As a teacher, I value the feedback from my students. I believe that teaching is a dialogue between a teacher and students, and I use feedback from students to improve my own teaching skills. This year I have been teaching tutorials in a class of Programming Languages. Every two weeks I delivered a tutorial to around 100 students, either on supplemental knowledge for their projects or reviewing their assignments or exams. Each tutorial was run as two sessions in two weeks where students could choose any one to attend. Though students typically only chose to attend the first one mainly, I saw this as a great opportunity to improve my teaching from students’ feedback. At the very beginning of each tutorial, I encouraged students to tell me what they thought about my teaching, either in class or after class by email, and I tried to create an atmosphere of dialogue between students and me in the tutorial. After the first session, I collected the feedback I received from my students and made adjustments to the second session. Eventually, the result of this approach was increased attendance of both sessions, with more students choosing to attend the second one.

I enjoy teaching because I can share my passion for the computer with other people. My experiences of teaching in different settings have also rewarded me a lot. I believe that it takes a long journey to become a good teacher, and I would like to take various measures to refine my teaching skill, including taking professional training opportunities, learning from education theory, taking advantages of new technology, and exploiting local resources. For example, I participated in Teaching in Higher Education this year, a course given to senior Ph.D. students of the University of Toronto to learn different teaching strategies and methods from a variety of teaching situations. I have also volunteered to give lessons of scientific programming in a local study group. Having been a student for many years, I have understood the importance and value of a teacher’s role in a class, and hence I will integrate the ideas I have learned from these experiences to design innovative strategies for courses in computer science and engineering at the university level.

In particular, I am strongly interested in teaching courses in programming, distributed systems, linear algebra, discrete mathematics, algorithms, data structures, and other undergraduate courses. I am also well prepared to teach advanced graduate courses related to my research areas, such as distributed storage systems, big data processing systems, information and coding theory, and graduate seminars. In addition, I would like to expose undergraduate students to some research opportunities, giving them a real chance to feel the excitement of doing research.