Teaching Statement

Yuzhe Tang, yztang@gatech.edu

Teaching computer science is one of the major reasons that I have chosen to pursue an academic career. In my opinion, computer science is a pragmatic discipline with real-world influences. I wish to spark my students' interest in computer science by empowering them with skills to create pragmatic programs that will help them find better employment opportunities.

As a graduate student, I served as a teaching assistant for CS3310 (Operating System). My job was to design a course project on kernel programming. The teaching goal was to empower the students to understand how an operating system works and how to extend it rather than simply using it. Because my students were undergraduates and kernel module programming is known to be difficult, I decided to use a teaching method that my students can easily follow. First, I did an introductory lecture about the project overview. The lecture also included a live demonstration which gave them an idea of what was expected in the project deliverable. I presented a high-level process that students could follow to reach the goal. Then I held several follow-up sessions for Q&A. I prepared in advance a detailed step-by-step solution for possible crash screen problems that my students might encounter. During our interactive Q&A sessions, as expected, many students were troubled with crash screens on their experiment machines. Instead of giving out the solution directly, I asked them to make a prediction on which code line in their program could be causing the problems. Then, I walked them through the process to verify their prediction based on the step-by-step solution. By asking for a prediction, I wanted to empower my students with not only the knowledge but also the debugging skills that they can reuse in the future. At the end, the results of this project were surprisingly good: most students successfully delivered a working demo. From the student feedback at the semester's end, I learned that most students found the high-level instruction and Q&A sessions very helpful and effective. For example, one student said "if without the help from TA, I could have not be able to finish the experiment on time. The Q&A sessions clearly saved me a lot of time on debugging which would otherwise be very frustrating as I tried myself."

When I was a TA for a graduate-level course CS6675 (Advanced Internet Computing and Application Development), I was responsible for a research design project. I chose the big topic of cloud computing as cloud computing enables average users to do things bigger than their normal capacity, a computing paradigm that recently becomes popular in industry. And I wanted to empower my students with skills and hands-on experiences of programming with the cloud to improve their employment opportunities. To design the project, I followed a similar procedure with the kernel programming project by doing an in-class tutorial on a cloud computing platform (e.g., Emulab) with a step-by-step workflow. However since this time I was dealing with graduate students, I tended not to help the students with detailed programming problems, which they were requested to solve by themselves independently, but instead focused on advising my students on research aspects of the project, such as choosing a research topic and analyzing experiment results. To help my students freely choose topics based on their interests, I gave them a list of potential options. I tried my best to make the description of each topic interesting by making connections to a real-world scenario. I also handed out a set of related papers and other resources in case my students were interested and willing to dig in deep to the topic. It turned out my students are interested in the projects and one group of students was even working with me to turn their project into a research paper.

Again, my teaching aims at empowering my students with not only class knowledge but also the learning method that they can apply the next time. In CS4420 (Database Implementation), I was a grading TA and one responsibility was to guide students through writing a 20-page technology review at the semester's end. Instead of simply setting the goal without telling them how to reach it, I designed a bi-weekly reading program, which required my students to read a subset of papers under one topic every two weeks. This allowed the students to gradually approach the final goal, a strategy that can be applied for learning in this program as well as in the future. Every two weeks, students would submit a write-up where they summarized each paper and compared and contrasted different papers under the same topic. This helped the students evaluate each paper. After the students finished the bi-weekly write-up, I hoped it would be much easier for them to write the 20-page final report. During the follow-up discussion with my students, I learned that many students appreciated this reading program, as they had found the method was very helpful for digesting the new knowledge. One student said "during the final writing, I can consolidate the knowledge learned through the weekly reading."

As I stated above, I focus on empowering my students with computer science knowledge and skills, such as to extend OS, cloud computing, predicting and fixing bugs. Towards the goal, my general teaching method is to motivate my students by sparking their curiosity and interests in the class knowledge, and to guide them through the learning process in a top-down fashion by first showing the overview and outcome and then working out the details. I would apply this teaching philosophy to new classes that I can teach in the future, such as databases, operating systems, security, distributed systems, cloud computing and any undergraduate-level CS core classes.