

Teaching Statement

Tingxin Yan

University of Massachusetts Amherst
yan@cs.umass.edu

During the past ten years as a student in computer science, many great teachers have influenced me with not only knowledge and skills, but more importantly their curiosity and passion for computer science. Now, I am pursuing the opportunity to become a teacher to share my own knowledge, curiosity, and innovative spirit with the next-generation researchers and engineers. My experiences on course design, lecturing, and mentoring students have prepared me to teach a variety of courses and reaffirmed my desire to be a good teacher.

Teaching Experience

Much of my teaching experience has come from being a teaching assistant for CS377, the undergraduate-level operating system course. CS377 is regarded as one of the most challenging courses and always attracts best system programmers at UMASS as well as several other nearby institutions such as Smith college and Mt. Holyoke College. The most significant component of CS377 is its highly-demanding course project, which takes 50% of the course credits. As a teaching assistant, my major responsibility was to help students complete their course project successfully. More specifically, my work included the following three aspects. My first job was to re-design the course project. Starting from Spring 2011, we decided to build the course project based on Berkeley's Nachos OS¹. The Nachos OS is an Unix-like operating system with a significant portion left for students to implement. We tailored the Nachos OS to the expected skillset of our students, and designed three assignments corresponding to three core components of an operating system. Our project decreased the workload from initial Nachos OS, and was more focused on OS concepts rather than large amount of coding. This experience taught me how to design useful assignments where fundamental concepts can be tightly associated with concrete problems. My second job was to give weekly lectures on project-related topics. I used the lecture time to introduce the project assignments, explain the core OS concepts that are related to the project, and teach programming skills that are required to the project. This experience taught me a lot about crafting interesting lectures and presenting them in an engaging way. My third job was to host weekly lab sessions where students work on their project assignments while I was available to answer their questions. The lab sessions allowed me to find common problems faced by students, and individually help students solve their questions. While lecturing gave me experience of teaching a class of 30-40 students, the lab sessions gave me one-on-one mentoring experience.

Teaching Philosophy

The core of my teaching philosophy is to provide an enjoyable and manageable learning process to students. I believe that students, rather than lecturers, are the center of a course. On one hand, to encourage students' curiosity and passion for the course, teachers should present the subject matter in an interesting and engaging manner. On the other hand, to help students become self-motivated in a course, teachers should also guide students to manage their learning process, and give active feedback to students. More specifically, I am interested in pursuing the following pedagogical efforts:

Encourage active learning. Students need to get involved in classes. Instead of one-way communication, I believe lectures with interactions are more efficient in stimulating students' interests. When I was teaching CS377, I used several methods to encourage students' interactions, such as white board programming, group discussions, and quiz questions. These interactions encourage students to crystallize their thoughts and spark new ideas. For the lecturer, discussions are also helpful to evaluate students and solve their common problems.

Protect creativity. One of the most exciting things in teaching is to discover the creativity of students. Some of the students are much more creative than I would have expected, and they have their own way of accomplishing

¹Nachos OS: <http://www.cs.berkeley.edu/~kubitron/courses/cs162-F05/Nachos/walk/walk.html>

projects, manage time, and so on. I believe teachers should give more freedom to these students, and the evaluation process should be changed accordingly.

Add some humor. Although computer science is serious, the way of presenting it can be humorous. Adding some humor to the introduction of a course can increase students' interests, and some humorous analogies in slides can better explain the concepts that are difficult to understand otherwise. When I was teaching CS377, I carefully made slides to contain some humor in explaining the course project requirement, and I also used some funny icons on the course webpage. Students did notice these details, and the positive feedback I received from students was far more than I had expected.

Help students gain steady progress. Steady progress is very helpful to motivate students and increase their confidence on the subject. When I was teaching CS377, I have seen several students getting more motivated to the course after they received positive feedback for previous assignments. To help students progress steadily, teachers need to carefully design the course assignments and projects with multiple milestones, and introduce students to the subject gradually. For instance, we divided the CS377 course project into three components and designed a few milestones for each component. By doing this, students had clear goals of when to complete each of them. After students finished all components, they were amazed that they actually implemented a working operating system despite the fact that they felt it to be a daunting possibility at the beginning of the semester.

Solve problems at their early stage. When students fail to follow course materials, teachers should not wait until the last minute to help them. One useful tip that I learned from teaching CS377 is to ask students to submit design documents one week ahead of their actual implementation. It not only urged students to start working on project assignments early, but also gave me enough time to find their problems and address them during lab sessions.

Teaching Style

My teaching style is a combination of lectures, demonstrations, and hands on projects that present course materials in various perspectives. I will incorporate white board lecturing, PowerPoint slides, and real system demonstration into my teaching. From my experience with teaching CS377, I learnt that each of these pedagogical techniques has their own pros and cons, and a combination of them would provide an ideal teaching performance. The assignments and projects in my classes will be a real-world challenge but carefully divided into a set of self-contained problems. Many courses, such as operating systems, networks, and mobile computing, provide excellent opportunities for designing interesting hands on projects that not only reinforce course material but provide students with the skills they need for their future jobs.

Teaching Interests

I would like to teach conventional courses such as operating systems, networks and distributed systems. I am also excited to apply my research to teaching. I am prepared to design several graduate-level seminars that are closely related to my research, such as mobile computing, mobile sensing and crowdsourcing. I envision that students could exchange and crystallize ideas through extensive paper reading, and gain some hands-on experience through projects that reveal real world challenges. I also plan to involve undergraduate students to my research through taking my courses. I am prepared to offer an undergraduate-level mobile programming course, which could give students a flavor of programming on popular mobile platforms such as iPhones and Android Phones, and provide students opportunities to be involved in research projects and gain experience of solving real-world problems.