Teaching Statement

Ryan P. Huang

Teaching has a profound impact in society. I personally benefit from numerous excellent professors who taught me. This motivates me to pass on what I have learned to benefit others. My goal in teaching is to cultivate interests in essential knowledge among students, help students acquire solid skills, and inspire students to use the skills to tackle real problems. Through past experiences of teaching undergraduate and graduate courses, as well as through mentoring students, I have developed a keen interest and concrete philosophy in teaching. The opportunity of being able to continue teaching is a key attraction of an academic career to me.

Teaching Experiences. I started to realize the power and joy of teaching when I was a senior at Peking University. I was given an opportunity to be a teaching assistant for the Introduction to Computation course. The course is designed for freshman in CS to cover basic programming and core algorithms. As the only undergraduate among the TAs for the course, I was initially worried about my experience. But then I found that starting from a rudimentary context helps me explain some elusive materials (e.g., dynamic programming, recursion) more effectively. I enjoyed helping students who have difficulties with the assignments. I liked encouraging students who work out the solution to further improve (e.g., “That’s great. But can you think of an $O(n)$ solution?”). The teaching also deepens my own understandings of the course materials, allowing me to share them with students in a better way.

During my Ph.D. at UCSD, I have sought out more opportunities for teaching. I was TA for the undergraduate Operating System course (CSE 120) twice, and the graduate Operating System course (CSE 221) once. OS is a difficult course for many students as it encompasses a broad range of principles. Learning OS requires not only conceptual understandings, but also intensive engineering to relate principles to implementations.

CSE 120 aims to expose students to principles via regular lectures, and implementations via course project on a teaching OS (Nachos). This creates a heavy workload on students. As a TA, I feel responsible to provide hands-on help to students as much as possible: I organized weekly discussion sections to recite the concepts under the context of Nachos; broke the large project assignments into smaller modules to explain; offered high-level design hints as well as low-level tips; held office hours, and walked in the labs where students work on the assignments to answer specific questions. I learned that a right presentation helps students understand difficult concepts clearly. As an example, in the first time of teaching the course, I mainly talk using slides. In the second time, I changed to the basic blackboard approach to slowly walk through each important point: e.g., I would draw an anatomy of computer and gradually show the details of virtual memory.

These efforts tremendously helped students to grasp the materials and make progress in the projects. For example, a student said: “All of his discussion sections were always really good. He would dive into the material and explain piece by piece to connect it all together[...] I thought it was great that he was trying to explain the material from the very beginning and not just a piece of the project.”

As a graduate course, CSE 221 focuses on the research aspects of OS. It has intensive readings on both classic and recent OS-related research papers. Additionally, it has a project on OS performance measurement where students are encouraged to be creative. The teaching experience of this course is different compared to CSE 120: students in the class prefer discussions on ideas and critiques.
make sure to provide high-level feedback while keeping the discussions open, and encourage students to think critically.

Teaching Philosophy. The interactions with students not only add to my teaching experiences but also help shape my teaching philosophy around four aspects. First, teaching should be engaging. Engaging the students to participate makes learning active rather than passive. Moreover, it triggers curiosity so that students can extend the learning beyond classroom. Second, for many CS courses, teaching needs to be accompanied by hands-on elements such as discussion sections, design reviews, and in-lab advising. Otherwise, students would be struggling with the abstract, hard-to-understand concepts. Third, similar as my research philosophy, I would expose students to real-world development in CS. For example, there are many high-quality open-source projects available. Using them as references or course project basis can allow students to learn from examples. Introducing the latest industry experiences is another way for students to stay in tough with the real world. Forth, ensuring periodic feedback is indispensable in teaching. The feedback can be from students to teacher on the teaching methods and topics, as well as from teacher to students on homework, project, and career advise.

Mentoring. I view mentoring students as an intellectual exchange to provoke thought. While working on my research projects, I have the pleasure to mentor several junior Ph.D. students at UCSD. The mentoring was effective in helping junior Ph.D. students gain experiences on attacking a research problem, executing an idea, and expressing the research in writing. Besides, I advised Vincent Chen (now at Facebook) for his Master thesis on learning from imbalanced performance testing data. The advising was hands-on, including coming up with the project idea, organizing weekly project discussion, teaching necessary background, and helping with the thesis writing.

Through these mentoring experiences, I learned that it is crucial for advisors to understand students’ strength, choose projects that match the strength, and advise in a style that works best for the students. Besides, advisors should actively observe and get feedback from students. For example, I listened carefully when Vincent indicated that he got lost in the big picture of the project. We then had a long discussion to keep us on the same page.

Courses. As a professor, I would be qualified to teach operating system, distributed system, software engineering, programming language, algorithm and data structure, mobile development, and seminars on system research. In addition, I would like to contribute to diversity in CS education by helping women and minorities in class and graduate studies. This is an important problem in education that professors should participate to solve. I am excited about the opportunities to interact with a diverse set of students and train future leaders in CS.