Teaching Statement

Ang Chen

I am passionate about teaching, and the opportunity to teach is an important factor that motivates me to pursue an academic career. My goal in teaching is not only to convey tried-and-true knowledge to students, but more importantly, to help them discover the wonders of computer science for themselves and to cultivate in them a deep interest in intellectual endeavors.

**Teaching:** I have been a TA for three courses: a) Scalable and Cloud Computing (NETS212), a sophomore-level course at Penn that teaches the basics of cloud computing and distributed systems, b) Software Systems (CIS505), a core course in the graduate program at Penn that covers fundamental concepts in operating systems and distributed systems, and c) Internet Infrastructure and Protocols (COMP5311), a graduate course at the Hong Kong Polytechnic University that focuses on the Internet architecture and its protocol suite. In each of those courses, I have interacted closely with students to provide guidance to them, and I have assisted the main instructors in designing homework sets and course projects. In the latter two courses, I have also given guest lectures.

I have learned several valuable lessons from my experience as TA and guest lecturer. First, when TAing the Scalable and Cloud Computing course, I observed that undergraduate students are interested in new, “cool” technology; therefore, a good way of engaging them is to create homework and projects that allow them to explore the latest technology. For instance, in the course project, the main instructor asked students to build their own mini-Facebook. The students were very enthusiastic when doing this project because it involved a variety of popular tools and systems, including Node.js, Hadoop MapReduce, and Amazon DynamoDB. I plan to use a similar way of engaging students when teaching my classes in the future.

Second, I have learned that students have different backgrounds, and that it is important to tailor the teaching style accordingly. When I was TAing Scalable and Cloud Computing – part of an inter-disciplinary undergraduate program at Penn – there were students from 16 different majors. This heterogeneous student body created an interesting challenge: since students had very different levels of exposure to computer science, I needed to be prepared to answer a wide range of questions, from technical questions about the internals of MapReduce (e.g., from computer science students) to questions about the latest computer technology and business trends (e.g., from students in the business school). Since not all the students had prior programming experience, I also gave them hands-on help as much as possible, by offering introductory lab sessions on JavaScript and interactive debugging sessions, as well as writing example framework code in Node.js.

Third, it is important to not only teach students concepts in the abstract, but also to give them opportunities where they can put their knowledge into practice. For instance, when TAing Software Systems, I found that students developed a much deeper understanding of leader election protocols after they built a distributed chat system with this feature in a course project. I also found that many students value hands-on experiences where they could get involved in open-source projects, or release their own project code.

Fourth, integrating recent research results into special lectures or course projects can be a good way of capturing students’ interest. When TAing Internet Infrastructure and Protocols, I incorporated research elements in the design of the course projects. I gave a tour on a recent research project called TraceNET and asked students to study the research ideas involved, provide critiques, and report results from experiments with the tool. I was impressed by the students’ enthusiasm in reading the research paper, and by their interesting critiques and ideas.

Drawing from my past experience, I summarize my teaching philosophy as follows. First, I believe that teaching should go from practical examples to theoretical concepts, and then back to practice. Computer science has many practical impacts on our society, and many of its concepts are also motivated by real-world problems. When possible, it helps to start with everyday examples, and then go deeper into the theoretical concepts or abstractions. When I offered my guest lecture on deadlock prevention and avoidance, I used many small examples to illustrate the key concepts and algorithms, not only on the slides but also by drawing on the board. I found that concrete examples can make otherwise difficult concepts much easier to grasp. Eventually, the students should also be able to apply their knowledge to solve real-world challenges, for instance, by designing and building system prototypes for a particular application. Second, I believe that
research – the process of discovery – should be better integrated with classroom education. I plan to do this in the form of giving special lectures on recent research advances, and by encouraging students to read classic papers on a particular topic. Third, as an instructor, I want to be constantly learning about effective ways of communication – for instance, by finding new illustrations of a well-known concept – so that my teaching skills will improve over time. I plan to achieve this by regularly gathering student feedback – for instance, from course evaluations or in-class surveys – and by integrating their feedback in future course offerings.

I am qualified to teach courses in computer networks, distributed systems, operating systems, and security. I am also interested in developing a course on the intersection of networking and databases. Building on my research background, I plan to cover topics such as data provenance for large-scale networks, designing distributed systems for “big data” processing, using networking hardware such as NetFPGAs for data analytics. I expect such a course to be useful to graduate students who are interested in recent developments in these areas, as well as to advanced undergraduate students.

**Mentoring:** I have had very rewarding experiences mentoring younger students, including Masters students (Akshay Sriraman, Robert Dimaiolo, Chirag Shah) and a junior Ph.D. student (Max Demoulin). I am very proud to see my mentees’ growth and every bit of their success. In the SplitStack project, I provided guidance to Akshay several times a week on experimental design and formulation of research arguments; this mentoring relation resulted in a HotNets publication. The two other Masters students I have been mentoring, Robert and Chirag, have also been working on the SplitStack project. I mentor them weekly by outlining possible ways of going forward in a research problem when the picture was not clear, as well as by steering them to a different route when the original plan did not go as expected. I am very happy that Akshay and Chirag were able to use their research experience in a recent job search. I have also started working with Max, a first-year Ph.D. student. We have regular meetings to discuss his research on the SplitStack controller, including designing scheduling algorithms, architecting the system, and also designing experiments. With Robert, Chirag, and Max, we are aiming at a SIGCOMM submission, which is in January 2017.