Teaching Statement

Chenyuan Wu

Helping students understand difficult concepts they are struggling with and witnessing their "aha" moments of understanding is the most rewarding part of an academic career. During my graduate school, I was fortunate to work as a teaching assistant and mentor to junior students, fostering both their intellectual growth and research interests. These experiences deepened my commitment to education and mentorship.

Teaching experiences. In 2019, our school launched the online Master of Computer and Information Technology (MCIT) program, a large-scale online master's degree program in Computer Science. With over 2,000 students, the program mainly serves mid-career professionals seeking to transition into computer science and advance their technical skills. I contributed to the program twice, serving as both a course designer and teaching assistant (TA) for the new Networked Systems course, which covers fundamental concepts in computer networks including routing, transport protocols, and application layer protocols. Since the program is designed for students with minimal engineering backgrounds, it placed additional emphasis on managing course difficulty and ensuring clarity in the content.

As a course designer, I was in charge of designing and implementing the second and most challenging programming assignment: a search engine based on distributed hash tables. This assignment requires students to apply various concepts they have learned in the course such as overlay networks and the Chord protocol. To make the assignment more manageable, I divided it into two milestones: (1) a basic Chord implementation where the ring stabilization protocol can work, testing student's mastery of protocol correctness, and (2) a complete search engine implementation with Chord finger tables, assessing their ability to optimize protocol efficiency and understand concepts like inverted list publishing. In addition to writing a demonstrative implementation of the two milestones, I also created a comprehensive assignment guide for students and future TAs, including a project introduction, key concepts explanation, ns3 code base walk through, desired system behaviors, milestone instructions and opportunities for extra credits.

In the following semester, the course was officially launched. Then, as a TA, I actively took part in all aspects of the course: holding recitations for programming assignments, holding office hours, answering students' questions on online forums, and grading projects, mid-terms, and final exams. Since this course covers a lot of distributed protocols, plain algorithmic description might be hard for students to understand. Therefore, in my recitations and office hours, I drew diagrams on the whiteboard, illustrating how each node performs at each step of the protocol and inspiring students to think why the protocol needs to be designed in this way. My guidance not only helped students understand the concepts, but also developed their troubleshooting skills in distributed systems programming assignments.

Overall, I gained hands-on experience in developing and running a massive course from scratch.

Teaching philosophy. I believe in two core principles for teaching computer science, especially for systems courses. First, collaborative and inquiry-based learning. Simply providing the solution, proof, or algorithm to a problem without discussing all other (incorrect) possibilities does not foster true understanding. I prefer to become wondrous and inquisitive right alongside the students. For instance, when teaching the complex Paxos protocol, instead of proving that it requires $2f+1$ nodes, I discussed the case with a single node. In collaboration with students, we found scenarios where it does not work. We then increased the number of nodes to two, then to $f+1$, and finally to $2f+1$ while finding different counterexamples for each case. This way of exploring the problem domain makes complex problems more understandable.

Second, getting hands dirty is essential. While course lectures can teach students about the theory, no amount of reading or listening to lectures can replace actually implementing and debugging the system. Programming assignments reinforce concepts, as corner cases inevitably surface during development. I design my programming assignments in a way that ensures a student with an acceptable understanding of the material can receive an acceptable grade, but getting an exceptional grade requires an exceptional understanding of the material beyond the lectures. This means designing programming assignments that are manageable by a particularly busy student, but can also be explored further by a highly motivated student. In systems courses, such deeper exploration could be achieved through system performance optimization or adding extra functionalities for the system.

Teaching interests. My experience and expertise has prepared me to teach a broad range of courses. I am capable of teaching students the internals of blockchain systems, distributed systems, database systems, networked systems, operating systems, and big data. I am especially excited to teach the advanced topics in blockchain systems, along with the practical aspects of building such systems.

Mentoring. During my graduate study, I had the privilege of mentoring four students for research: two junior graduate students (Bhavana Mehta and Heena Nagda) and two undergraduates (Jared Asch and Haoyun Qin). My goal as a mentor is to train students to become independent researchers in computer science. Since passion and perseverance are key to becoming a successful researcher, my mentoring philosophy is to nurture students' passions and enlighten them about the importance of perseverance when necessary.

The students I mentored were passionate about different topics. Some enjoyed inventing distributed algorithms, while others were more interested in implementing and optimizing systems. For junior students who just get started in research, my approach is to offer them a variety of research avenues, and encourage them to select the topics that truly attract them. For more senior students, I encourage them to independently propose interesting topics and problems in their fields. In practice, I found this is a good way to cultivate interests and train students at different milestones. Perseverance is especially crucial in systems research, where the process of developing a novel system is often lengthy and filled with difficult moments. In these moments, helping students recognize the impact of their work and reminding them of the big pictures are as important as assisting with technical details.

One notable outcome of my mentoring is Haoyun Qin. When we began working together, Haoyun had just completed his first semester as an undergraduate at Penn and had no prior knowledge of distributed systems. After two years of mentorship, Haoyun co-authored four papers with me, all published in top conferences like VLDB and NSDI. Notably, he published a first-author demo paper on distributed protocols in VLDB 2024 (a top database conference), won Computing Research Association Outstanding Undergraduate Researcher Award, and has now decided to pursue a Ph.D. journey in the United States.