# Teaching Statement

Micah Sherr

Good instruction is an art and a multifaceted skill, requiring the teacher to be part orator, visual designer, moderator, entertainer, and above all, educator. Teaching requires the ability to synthesize complex and abstract concepts into a format that best conveys the material to students with diverse educational backgrounds and learning styles.

Despite its challenges, teaching provides benefits not found in any profession outside of the classroom. Participating in moments when students suddenly comprehend a difficult concept or become excited about the subject matter is a rewarding "perk" that is unique to teaching. My desire to teach has been the impetus of continuing my education and pursuing a career in academia. Not only does teaching impart technical concepts and skills, it is a valuable means to engage students in research, train future scientists and engineers, and discover future collaborators.

My interest in teaching predates my graduate career. While working at Columbia University, I volunteered for two semesters as a science instructor for the University's Jobs and Educational Empowerment Project. In this role, I designed the science curricula for students earning their GED (high school equivalency) certification. I presented weekly lectures in biology, chemistry, and astronomy. The GED primarily tests *scientific literacy* – the ability to interpret written descriptions of scientific phenomena – as opposed to collegiate classes which stress in-depth understanding of course material. Although there are significant differences in content and structure between GED courses and university classes, my experiences as a GED instructor allowed me to develop my teaching style and increase my comfort-level at the head of the classroom.

At Penn, doctoral students are required to complete a two-semester teaching practicum in which they serve as teaching assistants (TAs). I was a TA for the undergraduate computer operating systems (CIS380) and the graduate distributed systems (CIS505) courses. Although my responsibilities were mostly limited to holding office hours, preparing projects, and grading homework, my role as a TA gave me an appreciation for the level of preparation required to communicate advanced topics. Additionally, opportunities to guest lecture in both courses allowed me to further hone my skills presenting material in a classroom setting.

After completing the teaching practicum, I actively sought out additional teaching opportunities. For four semesters, I served as an instructor for the undergraduate operating systems (OS) lab course (CIS381), a required course for students who pursue the Bachelors of Engineering degree. Prior to my teaching the class, the course consisted of a large programming assignment and was commonly perceived as an extension to the operating systems lecture course. I redesigned the lab to include a teaching component alongside the class projects and homework assignments. I crafted a syllabus that covered advanced operating systems topics, introduced weekly tutorials, maintained class discussion boards, and supervised teaching assistants. I also designed two semester-long projects: (i) the implementation of a full-featured UNIX shell that supports background processes, job control, asynchronous job notification, piping, and input and output redirection; and (ii) a toy operating system featuring a multi-tiered preemptive scheduler, a FAT-based filesystem, a simplified shell, and a memory heap.

At its peak, the operating systems lab course had an enrollment of 65 students. Effectively communicating new material to a packed classroom can be a difficult task. I have found that an effective method of teaching complex topics is to take a top-down approach, presenting figures and diagrams that visualize core concepts before describing low-level details. To maintain student interest, my lectures are very interactive. Many operating systems problems have multiple possible solutions (for example, there are many ways to implement a filesystem). Before presenting standard solutions, I ask students to craft their own approaches during class. Often, classroom discussions lead directly to the correct solution. I have found that such a teaching style not only conveys information in an easy-to-grasp manner, it also highlights the rationale behind particular design choices and technological approaches. Mid-semester teaching evaluations indicated that students appreciated my teaching style and found the weekly tutorial sessions exceedingly helpful. Although attendance at the tutorials was not mandatory during the first two semesters that I taught the course, students attended the

sessions in large numbers. Due to the success of the lab lectures, the computer science department formalized the tutorial sessions, assigning them a weekly timeslot and making attendance mandatory. As a result of my work in the class, I was nominated by the department for a Graduate Fellowship for Teaching Excellence.

In addition to teaching in the classroom environment, I have also supervised several undergraduate senior design projects. These courses, structured as independent studies, introduce students to computer science research. During my own undergraduate studies, my senior design project helped spur my desire to participate in research. As a graduate student and later as a postdoctoral researcher, I volunteered to supervise these projects to try to convey this same level of excitement to my students.

There are significant differences between completing a class assignment and conducting research: the former is engineered to have an obtainable solution, while the latter does not necessarily result in a positive outcome. When mentoring undergraduates interested in research, I teach the students how to apply the scientific method to computer science problems. I stress the importance of constructing testable hypotheses, defining pertinent metrics, and designing useful experiments. The undergraduate projects have led to a publication at a top-tier security conference, and two additional papers are currently under review at a top academic database venue. Several former students have decided to enroll in graduate school, including two who are pursuing Ph.D.s in computer science at Harvard University and the University of California at Berkeley.

In the future, I would be delighted to teach any introductory-level undergraduate computer science class. I am particularly interested in teaching advanced undergraduate and graduate-level classes in networking, software security, databases, and operating systems, as these areas closely match my research interests. I will leverage my past experiences when designing these classes and will incorporate interesting and challenging assignments into the syllabi. In particular, I will encourage participation in research both as semester projects and as independent studies. Additionally, I am particularly interested in leading graduate-level seminars on software vulnerabilities, privacy-preserving technologies, and voting systems. These seminars will be structured to allow students to present both seminal papers from the literature as well as more recent work.