Xin Zhang — Teaching Statement

Computer science has been one of the fastest-growing fields in the past two decades. As a result, there is an increasing demand for qualified graduates in both industry and academia. To meet this dire need, I believe the mission of a computer science academic is to produce both software engineers that can solve real-world problems and researchers that can expand the frontiers of current technologies.

My experiences in teaching and student mentoring over the past five years have prepared me well for this mission: I have served as a teaching assistant for both undergraduate-level and graduate-level courses, contributed to course development in a massive online Master's program, and mentored both undergraduate and graduate students in research. Based on the different career paths that a student may take, I will apply different teaching methods that are inspired by these experiences.

Classroom Teaching. I am most interested in teaching courses related to my area of research—programming languages, compilers, and formal methods—as well as operating systems, databases, and algorithms. During my graduate study at Georgia Tech, I have served as a teaching assistant for an undergraduate course, *Introduction to Databases*, and a graduate course, *Software Analysis and Testing*.

Since most students will pursue jobs in industry, any course I teach will involve heavy hands-on experience in order to prepare them for solving real-world problems, For instance, in programming-related courses, I will include coding tasks, both in the form of weekly assignments and semester-long projects.

Hands-on experiences can help students gain a deeper understanding and longer-term memory of the course material. One example is from my own experience in serving as a teaching assistant for *Software Analysis and Testing*, a graduate course with over 40 students. It includes several lectures on foundational theories of program analysis that many students initially had trouble comprehending due to heavy formalism. However, the concepts quickly became clear to the students after they worked out the related coding assignments. I had a similar experience for my undergraduate study: the courses with hands-on exercises were the ones from which I benefited the most in my future career.

Hands-on experiences also directly prepare students for industry jobs by allowing them to solve problems that resemble real-world situations. In particular, I will employ semester-long, group-based course projects, which have several benefits. First of all, they can be modeled as real problems in industry, thereby directly training students in technical skills required for industry jobs. Moreover, working in groups enables students to learn from each other, thereby facilitating the process of learning. Last but not least, these projects inculcate not just technical skills, but also "soft" skills in presentation, communication, and collaboration. One example was from my experience in serving as a teaching assistant for the undergraduate course *Introduction to Databases*, where the students came from different majors. The students were asked to form groups of three to build a car rental system that resembles the one used by Zipcar. Interestingly, the group that performed best was not formed of computer science students who were the most technically skilled, but industrial and systems engineering students who worked together better as a team. The project effectively assessed the complete skillset of a student as a software engineer. Moreover, it raised their awareness of the importance of soft skills and helped them improve on these skills.

Online Education. Massive Open Online Courses (MOOCs) have emerged as a promising way to train a large number of qualified computer engineers at low cost. In 2013, Georgia Tech launched the first massive online Master's program. I helped develop one of the courses in the program—the online version of *Software Analysis and Testing*. My responsibilities included designing key parts of the lectures, assignments, and exams. Through my experience, I have identified three central challenges in online education and thought of their corresponding solutions, which I outline below.

Scalability. The enrollment size of an online course is much larger compared to its on-campus counterpart. While the number of instructors may not increase substantially, the number of required graders tends to grow linearly with the number of students. Since graders are typically on-campus students, the rapidly growing online enrollment poses a serious challenge for hiring qualified graders. Recently, program analysis and synthesis techniques in my area of research have shown promise in providing automatic real-time feedback and automating the grading of programming assignments. I plan to investigate such techniques as a part of my research and apply them to online education.

Diversity. In the absence of of physical constraints such as locations and schedules, the online student body tends to be even more diverse compared to the student body on campus. For instance, the average age of students in Georgia Tech's online Master's program is 10 years higher compared to that of students on campus. To teach a class with such diversity, my general principle is that the instructor should be aware of the background of different individuals and adjust the course delivery accordingly. The former can be achieved by conducting surveys on the students' background upfront and surveys on course content throughout the course. Although it is difficult to customize the course delivery for each individual student due to the large enrollment size, the instructor should at least ensure that it does not exclude any group of students.

Communication. Online students may find it more challenging to build personal connections with their classmates and the instructor due to vast physical distance. Again, I believe technology can help address this problem. Georgia Tech has leveraged multiple online platforms including Piazza and Google Plus for students and instructors to communicate efficiently. The discussions on these platforms are as engaging as the ones in classrooms, if not more. To build a more personal rapport with the students, my teaching assistant colleagues in *Software Analysis and Testing* held weekly online office hours via video broadcasting.

Undergraduate Mentoring. Besides preparing students that aim for industry jobs, I will also nurture students who wish to pursue a research career. Most undergraduates form a clear idea about the career choice only in later stages of their study. To facilitate this process and identify students that are interested in research, I will add optional content to the courses I teach, such as research papers expanding upon the regular course content and even open research problems. After identifying these students, I will either work with them myself or recommend them to other faculty members who are better matches in terms of research interest. I had the opportunities to work with four undergraduates whom I connected through similar processes. Each of them was assigned a well-defined problem in a larger project that I was working on. Every week, we would hold a project meeting and discuss both the individual problems that they were solving and the overall progress of the whole project. Through our interaction, all students not only improved their technical skills but also formed a better understanding about the research career.

Graduate Advising. One of the most fulfilling experiences for a faculty at a research university is to advise graduate students in research. During my graduate years, I have worked with two junior PhD students and one MS student, which led to three research papers published at OOPSLA'16, AAAI'16, and CP'16 respectively. I made two key observations about student advising through our interaction. First, it is important for an advisor to work closely with advisees in early stages of research. For most novice researchers, the fastest way to learn about research is to observe how other researchers conduct it. It requires the advisor to have intensive discussions with the advisees and constantly give constructive feedback. I applied such methods for all three students. For instance, I spent one month with one student to produce three pages of math proofs, which involved ten iterations of feedback and revision. Though we spent a significant amount of time on what seemed simple, the student truly mastered the skills for solving similar problems after this experience. Second, it is important for the student and advisor to form a healthy relationship with mutual respect. Students are most productive when they are motivated. Such motivation does not solely depend on the students' interest in the topic, but also how comfortable they feel when working with the advisor. A stressful relationship can completely demotivate the students. Guided by this belief, my role in our interaction was more like a mentor rather than a supervisor. Even when there were times where the solutions proposed by the students were not viable, I did not dismiss them quickly. Instead, I analyzed the flaws of these solutions with the students and they eventually converged into suitable ones. In this way, the students maintained ownership of their projects and stayed motivated throughout the process.

In summary, I am passionate about both teaching (in classrooms or online) and mentoring students in research. With my teaching experience and research background, I firmly believe that I can cultivate a classroom environment that prepares students effectively for different career paths, and create a productive advisor-advisee relationship that motivates students in advancing the frontiers of knowledge.