Pranav Garg — Teaching Statement

I consider teaching to be a very noble profession, where a good teacher literally shapes the future of his or her students by providing them direction and harnessing their potential. At the same time, interaction with young students offers a unique opportunity to the teachers to continuously learn and grow. These are two aspects of the teaching profession that I greatly admire and are the primary reasons for my pursuing an academic career.

Teaching Approach and Experiences. My teaching philosophy has been shaped by my teaching experiences as well as my experiences as a student. Of all the good teachers I have had the opportunity of studying from, I learned the most in classes that were very interactive and discussion-oriented, taught by teachers who were extremely passionate about their subjects and whose passion and engaging style would always interest and inspire the students. I have always modeled my teaching on similar lines. At UIUC, I was one of the teaching assistants for CS428, the undergraduate course on software engineering, which is extremely popular amongst the senior undergraduate students. This course had over 150 students and apart from managing many of the logistics of the course, including homework assignments, moderating the online class forum, designing and grading exams, and conducting exam review sessions, I was also responsible for mentoring over 50 students on their group projects. Besides, I also gave a guest lecture and led several in-class problem solving sessions. I have also had teaching experiences at my undergraduate school, IIT Kanpur in India, when I was one of the instructors in a summer school on data structures and algorithms course.

My teaching approach focuses on incorporating hands-on experience and real-world problems in the classroom environment. As a teacher, I put a lot of emphasis on project work and assignments and I see them as very essential components that teach students how to apply the concepts taught in the class in practice. Moreover, explaining a new concept in the context of projects or assignments students are working on or by giving examples from the real-world that students can relate to helps them internalize the concepts and retain them for a longer period. For instance, in the project meetings with my students, I would often initiate short discussions about the concepts covered in the class, and give the students small thought exercises that would force them to apply those concepts in the context of their projects. As a simple example, in meetings after the class on software planning and risk analysis, I asked my teams to think about various risks to the viability of their projects they foresee and actions they plan to undertake to mitigate those risks. Just like projects and machine problems in my opinion are invaluable components of a course curriculum, in a theory course I feel there is no replacement for practice. I believe that students best understand complex proofs by going over them together step-by-step on a blackboard. Therefore, I prefer a mixture of presenting and working with the blackboard in my lectures on theoretical topics. In addition, I believe homeworks and practice problems are very effective for students to get a hang of the theory taught in the class, and I like to have them in my courses at regular intervals.

My teaching style is very interactive, and during my lectures, I regularly invite questions and involve students personally in discussions. My teaching experiences have taught me about the challenges that come with teaching large classes that have a huge diversity amongst the students. During my meetings with the project teams, as much as I enjoyed working with high achieving students, I would often pay more attention to students who were finding the course material challenging and would encourage them to take up more responsibilities in their projects. At the summer school in IIT Kanpur, I had a similar experience where many students attending the workshop came from diverse non computer science backgrounds. In such a scenario, I believe, it is very important to make sure that the high achievers do not lose their interest by a relatively slow pace of instruction. I made sure this was not the case by providing the high achieving students optional challenging problems to keep them motivated.

Teaching Interests. My expertise broadly lie in the areas of programming languages, formal methods and software engineering and I eagerly look forward to teaching undergraduate and graduate courses in these areas, including courses on programming language design, compilers, program analysis, automata theory and logics in computer science. I am also well acquainted with standard contents of undergraduate courses on discrete mathematics, data structures, algorithms and machine learning and given the need I would readily commit the effort to effectively teach these courses. At the graduate level, I would like to offer an advanced course on program verification that would include topics such as abstract interpretation, symbolic execution, deductive verification, decision procedures, program logics, and automated theorem proving. Applications of these verification techniques to building verified and secure systems is an area I primarily work on and I would be very interested in offering a hands-on, project-based course on this topic. As a graduate student, I found great pleasure in reading and understanding seminal or “classic” works in my area, and I would also love to lead reading groups or seminars that discuss seminal and cutting-edge research papers in programming languages and formal methods.