Teaching Statement
Shomir Wilson

My teaching philosophy is based on three foundational principles: collaborative learning, classroom engagement, and development of communication skills. I have developed these principles from my experience as a teacher and mentor of undergraduate and graduate students over the past nine years. I explain each principle below with examples from my experiences.

Teamwork is an essential skill for computer science students, who must develop the collaborative software development habits that are necessary for their future careers. In spring 2015 I co-taught Carnegie Mellon University's Natural Language Processing (NLP) course, which included a term project for teams of three to five students. Teams worked competitively to build question answering and question generation systems based on the contents of Wikipedia articles. The flexibility and scope of the assignment required students to combine their talents toward challenging goals that they could not reach individually. A student's grade on the project reflected a combination of their own contributions to the team and their team's performance, and teams with outstanding results were recognized with prizes at the end of the semester. Additionally, as a member of the Mobile Commerce Lab, I have supervised several undergraduate and masters-level researchers with important team roles in software development and user studies. I initiate these students with simple tasks on an existing code base, and then ask them to make design decisions of increasing importance. Through this experience, they learn to add their unique strengths to a collaborative effort. I intend to incorporate teamwork as a core element in future courses and mentoring, with classroom activities and projects that induce student cooperation. To encourage exceptionally talented undergraduate students, I will continue to seek out those who will benefit from joining collaborative research efforts.

Active engagement in the classroom experience maximizes the value of students' time spent there. For example, I concluded each of my NLP lectures with an interactive demo of the ideas or methods that I had introduced. This scaled well for the lecture size (approximately 60 students) while it maintained student interest, as I asked students to provide sample input and to predict how NLP tools would respond to it. This student interaction also provided me with feedback on my lecture: I was able to correct misunderstandings and explain key concepts in different ways. For example, some students' reactions to a word sense disambiguation demo showed me that they did not understand the limitations of a bag-of-words model, and I showed them input examples that required greater sophistication. As a teaching assistant at the University of Maryland, I led a discussion section for a course that introduced students to system programming concepts. I explained concepts from the lectures in greater detail, introduced new material, and led students through programming exercises to apply what they had learned. Students immediately saw the results of their efforts and could quickly receive assistance with conceptual and technical problems. Finding ways to break out of the traditional lecturing model is one of my priorities, for both small and large classes, to maximize students' gains from their time in the classroom.

Computer scientists must be able to communicate clearly about their work, and such communication is learned only with practice. The NLP course project required students to prepare a series of progress reports, which included a written proposal at the beginning of the project, an oral report at a meeting partway through, and a video report at the end. As a teaching assistant at Virginia Tech, I held a discussion section for Freshman Honors Seminar, a class that introduced college freshmen in the University Honors Program to the opportunities that could help them make the most of their time in college. From both of these experiences I observed the value of encouraging students to express themselves clearly and articulately, both in written and verbal contexts. Developing their communication skills helped them clearly demonstrate what they had learned and ask the right questions to further their understanding. In future teaching endeavors, I expect to emphasize the value of effective communication, for example through written technical reports and targeted feedback.

Looking forward, I am particularly interested in teaching courses on topics such as natural language processing, artificial intelligence, or usable privacy. Teaching NLP and mentoring undergraduate researchers are two activities that give me fresh perspectives on my research. For example,

preparing a lecture on discourse analysis led me to view lingual links (as described in my research statement) differently, leading to components of an NSF grant proposal co-written with two colleagues. I am also interested in teaching introductory-level courses, as I have found success with building students' enthusiasm for learning fundamental concepts in computer science. I will continue to look for opportunities to work with outreach programs such as the Intel International Science and Engineering Fair and FIRST Lego League, both of which I have volunteered for on multiple occasions. Such programs promote diversity in computer science, and they engage future computer scientists at crucial times in their primary and secondary education.