Nathaniel Ayewah

TEACHING STATEMENT

My teaching philosophy draws from my own experiences as a learner. I learn by putting ideas into practice, by synthesizing new ideas and by repeating the most important concepts. In my experiences teaching undergraduates how to program, or introducing advanced programming language concepts, I have tried to use these principles – practice, application, and repetition – to make learning more effective for my students. I help students put my lessons into practice by working through examples and inviting them to do the same. I also help students apply the lessons learned by discussing real moments in my career or research when the ideas have been helpful or relevant. And I repeat the important ideas often, sometimes using short, "easy" quizzes to reinforce the message. The reward for these exercises is observing students finally reach a clear understanding of the concepts I am teaching.

Most of my teaching experience has been as a teaching assistant, first for a sophomore level "Principles of Computer Science" course, and then for a senior level "Programming Language Technologies & Paradigms" course. In both courses, I was responsible for developing programming assignments, assisting students in their implementation, and grading the final product. I also taught lab sessions with 20 to 30 students. While teaching the sophomore level course, I came to appreciate the importance of constructing assignments that were relevant and engaging to keep students interested. At that level, many students were still deciding whether to commit to Computer Science, and my goal was to show that programming skills could enable them to tackle pertinent problems.

Prior to these teaching experiences, I was also an undergraduate tutor at the Learning Enhancement Center at Southern Methodist University. In these early days, I honed the skill of condensing complex concepts into simple statements and examples. Many of the fellow undergraduates I interacted with were taking calculus, computer science, or electrical engineering courses to satisfy a requirement, so motivation levels were not always high. Still, these students responded to my patient explanations, and this experience encouraged me to pursue more teaching opportunities in graduate school.

In graduate school, I had the opportunity to work on the Marmoset project. Marmoset is an award-winning system developed by the University of Maryland that enables students to submit programming assignments and receive instant feedback. One of our goals was to motivate students to write unit tests to validate the correctness of their implementations. To facilitate this, we partially hid some of the tests used to grade assignments, only indicating to students that they were failing some *release tests*. This encouraged students to think more deeply about the assignment's specification and creatively construct unit tests matching that specification. Marmoset also required instructors to create a fully functional canonical implementation before delivering the project to students, a practice that removed many bugs and flaws in programming assignment specifications.

Going forward, I am interested in teaching courses that introduce students to concepts in programming languages, software engineering, and human computer interaction. I am also interested in innovating new courses that explore the ways in which users interact with creativity support or productivity tools. Teaching courses close to my research interests creates opportunities to learn from students and gain interesting new perspectives as we put new ideas into practice together. I also believe it will be rewarding to introduce students to cutting-edge ideas, perhaps for the first time.