

## Teaching Statement—Kyle Burke

At the beginning of my teaching career, I unearthed a great joy in being responsible for educating students. Now I work very hard to give students every opportunity to grasp material, and find myself devoting more and more time and energy towards teaching. Since beginning to teach, I have also accumulated a number of policies, practices and philosophies which aid in communicating material to students.

Although I nearly avoided my initial opportunity to teach, once I had my first taste, I never looked back. In three years of college, I was a teaching assistant eight times; for two semesters I worked for both the math and computer science departments. During this time, I was unsatisfied with only grading for students. I allowed my duties to expand, first holding review sessions before exams, then holding “office hours” in my dorm where students could come and ask homework questions. Later, in my graduate career at Boston University, I continued this practice of extending my responsibilities. What began as a Teaching Fellowship matured into a year-long position as a lecturer, first teaching our department’s version of discrete math in the fall of 2007. Following that, I taught Geometric Algorithms in the spring—a testament to my ability, as my students from the fall semester petitioned for me to teach this class.

While working as a lecturer, I developed teaching policies for myself. Some are things I have learned while teaching, but most are a patchwork of the practices of teachers I have known. For example, I give my students simple quizzes at the beginning of some classes (determined randomly). This drives the students to review the last lecture’s notes before arriving, assuring that they are fresh on the topic when class begins. Additionally, I enforce a no-writing policy in my office to assure that students must both bring their own questions to office hours as well as consider the answers without simply scribbling down the work everyone else has brought. These policies help to set the stage, but are not harsh given the atmosphere I keep in the classroom.

Although policies can help a class run smoothly, it is not sufficient to evoke the correct attitude in the classroom. Before students can attentively focus on the course material, they must believe the course is run fairly. To this end, everything listed on the syllabus must be clear and well-updated. In this Internet age of “living” documents, the course webpage must contain all the relevant material for the course, laid out in simple fashion. In addition to the attitude of fairness, I have striven to expel competition from my classroom as best as possible. I refrain from curving grades based on averages and standard-deviations, instead explaining to students that they are being compared to all students I have seen before them. Combining this with the practice of allowing the students to discuss (amongst themselves) in-class questions during lecture has instilled great collaboration. This camaraderie and the fairness in the class lend to the crisp, positive attitude I strive for.

A big part of conquering both coding and math problems comes simply from confidence. Success in the classroom goes far in instilling this assurance. Although sometimes comments are too far off-track, I will tie anything relevant

back into the current material. It's always better to answer with "That's not exactly where we're going with this, but it does have an impact here..." than "No, that's not the answer." This is especially the case when the comment has relevance to more difficult material. Not only does this alleviate the discomfort of making an inane statement in class, but also gives curious students the chance to think about something beyond the standard material.

Although I have not taught a programming-centered course since being an undergrad, I hope to take on these responsibilities in the future. As a result of my training in Java, I would like to instill good object-oriented programming techniques in young students. (I don't, however, buy into the "objects first" mindset.) On the same track, I also look forward to the challenge of teaching an advanced object-oriented design course.

Naturally, I want to continue teaching theoretically-minded classes as well. In the spring of 2009, I will again apply my linear algebra background by teaching Geometric Algorithms here at Boston University. In the future, I hope to teach more discrete math, but also to apply my knowledge of game theory, randomized algorithms and parallel design in the classroom.

Students must be *shown* and not just *told*. With this in mind, I create an engaging environment with plenty of examples. My students quickly settle into a classroom where they will be challenged instead of threatened.