

# Teaching Statement

*Greg Bronevetsky*

I have a wide range of experience in the classroom, including working as a Teaching Assistant, designing and lecturing in my own seminar course, giving research presentations and one-on-one tutoring. These experiences have helped me to appreciate the complexities of helping students learn.

As a Teaching Assistant, I worked in a variety of Computer Science courses. “Computers and Programming” at Cornell University during the Fall of 2000 was a large introductory course where I was exposed to classroom-style teaching and course administration tasks. In teaching sections, I prepared material to present to students and answered questions they had about class material and assignments. The following semester, I worked on “Creating Web Documents” a large course for non-Computer Science majors. In this course, I managed other TAs and got more experience with grading, tutoring and course management. Finally, in the Spring of 2004 I was the TA for “Compiler Design for High-Performance Architectures”. This was a medium-sized (20 students) graduate course where my responsibilities included grading of assignments and administering several student research projects. Each project involved a preliminary (month-long) exploration of a research question, culminating in a report and an experimental evaluation. Further, I administered several class projects during other years that the class was offered, resulting in one such project being published at a major conference.

My most intense didactic experience was in teaching my own course: “Application-level Detection and Tolerance of Complex Faults” (<http://greg.bronevetsky.com/CS717FA2004>) with Keshav Pingali during the Fall of 2004. This seminar course focused on covering several decades of prior work on the detection and tolerance of complex system errors. As a result, I spent 12 lectures lecturing on various topics in the available literature and had the students pick their own papers to present in the remaining lectures. Each student was responsible for one or two lectures and had to do a novel research project of their choice, all of which I administered personally.

Since CS717 was a seminar course, its goal was to promote research. Our examination of the prior work on this topic was aimed at understanding the many untouched research areas in this field and use class projects to explore potentially promising ideas in these areas. In this sense, my job as instructor was to help students pursue new ideas and mentor them as their projects matured. However, another important thing I learned during the course was how to identify my audience and lecture to their background and needs. Since I had initially developed my lectures with no knowledge of what kinds of students would come to the course, as I learned the composition of the class, I adjusted my lectures to better align with their experience. In particular, given that most students were from Systems and Architecture, the Theory-heavy lectures had to be simplified. However, when I presented the same lectures at Theory discussion groups, I was able to concentrate their theoretical content to better align with this new audience’s background.

In addition to my formal classroom work, I have learned a great deal about education from giving presentations and from one-on-one tutoring. My presentation experience has spanned my entire career, including presentations at conferences, department seminars, informal discussion groups as well as in the classroom setting (in addition to my TA work).

As a result of my years of experience in education, I have learned a great deal about teaching. Furthermore, my wide range of research interests has given me a broad background in a number of areas of Computer Science. As a result, while I would prefer to teach undergraduate and graduate-level courses relating to High-Productivity Computing, Operating Systems, Fault Tolerance and Computer Architecture, I could also teach undergraduate course in other topics such as Programming Languages and Artificial Intelligence.