Statement of Teaching Qualifications and Approach

Brian T. Gold
Carnegie Mellon University

I view teaching as an extension of a fundamental aspect of research: the distillation of complicated and challenging concepts into their purest and most essential forms. In my own experience as a student, the most successful courses were taught by dynamic faculty who shared this view, engaging students in self discovery by combining classroom lectures, homework and lab assignments, and independent projects.

Course Offerings

I am most qualified to teach courses that cover areas related to my research in computer architecture. At an undergraduate level, this includes digital design, computer organization, and parallel/high-performance programming. At the graduate level, I am prepared to offer advanced courses in computer architecture, parallel and multiprocessor systems, and performance analysis.

Experience

At CMU, I have twice served as the teaching assistant for a course on multiprocessor computer architecture. In this role, I developed lab and homework assignments, organized and managed student projects, wrote exam questions, taught several lectures, held office hours, and provided grading support (including exams). In addition to the multiprocessor course, I provided support for research projects in several semesters of the graduate-level computer architecture course.

These courses, which typically enrolled 20 to 50 students, included a semester-long research project where students worked in groups of two or three. As teaching assistant, my role was to support the groups with regular technical discussions and assistance with tools and infrastructure. Because these courses target advanced undergraduates and early-career graduate students, we provided students with a hands-on introduction to current research tools and their application. In particular, we encouraged students to use *Flexus*, the timing-accurate, full-system simulator we developed for use in our own research group.

In addition to semester-long research projects, we employed shorter lab assignments to give students a deeper understanding of often-confusing subjects. For example, in the multiprocessor course, I created a cache-coherence lab where students develop shared-memory coherence protocols and verify them with a model checker. The lab required students to design their own coherence protocol and verify deadlock freedom and exclusivity properties. The results were extremely positive: by using model checking, students were able to quickly find bugs in their protocols, create solutions to common problems, and develop a deeper insight into coherence protocols than lectures can provide alone. I have published the materials for this lab, along with related course materials, on my web site.[1]

Approach

Every course has unique constraints: the methods and organization that work in a course with 20 graduate students do not often translate to freshman and sophomore courses with 200 students. In early undergraduate courses, I plan to supplement lectures with regular assignments that provide students with practice of the required material, while ensuring I get timely feedback about the progress and pace of the course. For advanced undergraduate and graduate courses, I plan to include semester-long research projects similar to what we have done at CMU. These projects are an excellent chance for students to engage in early-stage research, ideally bringing their independent interests to material from the course. My own undergraduate career was highlighted by research experience, and I plan to encourage students to take part in my group's research through independent study and similar offerings (e.g., NSF Research Experiences for Undergraduates).

---

[1] `http://www.ece.cmu.edu/~bgold/teaching`