Alexander J. Quinn ▪ Teaching Statement

In our department's *How To Do Research* seminar, professors take turns giving informal advice to incoming graduate students. On the topic of research writing, Professor Mike Hicks shared several memorable tips, among them this: *Make it your goal to teach your readers, rather than to simply impress them.* His point was about favoring clear communication and engagement over formality—that for an audience to go away fulfilled and changed, we must focus our energy on helping them understand.

Teaching, mentoring, and research are all about shared discovery

Classroom instruction is naturally a time of discovery as students are exposed to a new topic. My approach to teaching is to lead them down the path while always being there *on the path with them*. That means approaching each new concept almost as if we were learning it together. Ultimately, I measure my own success in terms of how well they can build on the knowledge to invent new ideas and communicate those ideas to others. While students are developing their foundation of theory and tools, they should be immersed in research as early as possible. As a research mentor, I will guide them to resources and help them select appropriately scoped problems that are likely to bear fruit, while they explore methods and technologies to solve the problem of the day. At all times, we will work from a basis of *shared responsibility* for the outcome. In the process, I aim to help them find the style of doing research that works for them and teach them to communicate our discoveries with increasingly wide audiences.

Use courses to engage students in research. Each time a student is excited by a course, it is an ideal bridge to get them involved in research. As a graduate student, three of my class projects were ultimately continued and published (two of them at CHI). Before that, two of my undergraduate instructors, Professor Steven Tanimoto and Professor Batya Friedman, mentored me on separate research projects, one resulting in a publication and the other a public software release. Their guidance and encouragement spurred my interest in graduate school, and also made me a staunch believer in undergraduate research.

Integrate active research into course material. I will look for opportunities to incorporate ideas and methods from current research into assignments within the context of course goals. I fondly remember an undergraduate artificial intelligence project created by Professor Dieter Fox, in which he shared with us his research code for mobile robot localization, but removed one section—a key algorithm. Our assignment, of course, was to fill it in. Knowing that what we were implementing was the leading edge—the best anyone knew how to do at the time—made the experience all the more fulfilling to us as students.

Promoting active learning in class

The ongoing debates around Massive Open Online Courses (MOOCs) has led me to consider new ways that technology could be used to structure classes. While in-person interaction is an indispensable part of education, MOOCs have cast fresh light on the idea of delivering lectures online. University classes typically involve some combination of traditional lecture (i.e., students listen while instructor explains), discussion, and activities. Since the lecture is primarily a passive learning activity, I would like to try prerecording it for students to watch outside of class. That would make more class time for active learning.

Teaching needs an *Edit-Compile-Test* cycle. Students may misunderstand material for a variety of reasons. To me, each misconception is like a bug that needs troubleshooting. I envision classes as debugging sessions where we use activities and discussions to root out misconceptions. Where possible, I would design exercises with deliberate pitfalls to detect when a student has misunderstood the content. The goal is for every student to contribute every day, and leave class with full mastery of the day's topic.

Give lectures a rewind button. Although shifting lectures to out-of-class time might seem like a compromise, I see it as a win-win proposition. A student watching at home could rewind to review a difficult spot, or pause to reconcile a point with the readings, potentially making the lecture *more* effective.

Classroom experience

I have learned about the finer points of instruction by assistant teaching with five faculty members—three at the University of Washington and two at the University of Maryland. In addition, I have been called upon to give guest lectures about human computation in four other classes. Before graduate school, I lived and worked in Japan, co-teaching English classes full-time at a public high school for two years.

At the University of Washington, I taught regular sections for a 2-part introductory programming sequence. The seasoned lecturers who led these classes showed me many of the tricks that they use to give a small-class experience even to students in such large classes. For example, Dr. Martin Dickey, kept a record of students who visited his office so he could follow up as needed. Later, I worked in an upper-level operating systems course. In addition to leading sections and grading, one of my tasks was to develop an all new project. I challenged the students to build a heap allocator—essentially a replacement for `malloc()`. The project description[1] had to be detailed, and yet leave enough freedom so that they would have to study and understand the underlying concepts, such as page replacement, page file management, and sub-page allocation. It was a challenging assignment and I worked especially closely with the students. Seeing them succeed with a combination of theory and practice made this a lasting inspiration for me.

Here at the University of Maryland, I supported the HCI course as a TA for two consecutive semesters. Students learned principles and methods for designing interactions. My primary role was to support them with assignments and projects that challenged them to analyze existing designs or invent new ones by blending design principles with creative thinking. Through my role in these courses, I developed a concrete method for maximizing the value of the feedback I gave on their work.

Balancing detailed feedback with consistent scoring

To grow and improve, students need both detailed feedback and healthy motivation. Although grades should never be seen as an end goal, they are nonetheless a powerful motivator, especially when students know that scores are consistent between students and directly tied to the course material. If accompanied by thoughtful comments, they can also ensure that every student receives a baseline amount of individual feedback. Balancing these two goals can be a challenge, since students expect there to be a direct link between the comments and the score. For the HCI assignments, it was often impractical to create a rubric in advance, since it was hard to predict in advance which areas students would have trouble with.

I developed a method for iteratively developing rubrics. As I checked the HCI assignments, I entered each comment into a spreadsheet along with a point value (penalty, bonus, or neutral). If I encountered the same issue, I would enter a code to refer to the first instance of it. Throughout the process, I checked that the relative proportions felt fair and consistent, and made adjustments, as needed. To enable personalizing the messages and more expressive scoring, I built an extension to Excel. The result was a set of scores with plenty of "spread" (i.e., entropy) and based on a rubric I knew was fair. For each assignment, my system sent students an email with detailed comments and a breakdown with their score.

I am proud of my solution for handling these assignments consistently and giving every student enough detailed praise and critique so that they could develop their skills. In addition to enhancing course learning, I believe fair and consistent scoring ultimately helps students discover areas where their strengths and passion intersect—an important end goal of every student's education.

Example courses

I would be pleased and qualified to teach undergraduate courses in introductory programming, human-computer interaction, computing ethics, and mobile programming. At the graduate level, I look forward to teaching courses in human computation, social computing, and information visualization.

---

[1] http://courses.cs.washington.edu/courses/cse451/02au/projects/proj3/proj3.html