Teaching statement

Raja R. Sambasivan

December 2011

I believe the most important goal of teaching is to instill in students a passion and long-term interest for the material. Inspired students will not always get a perfect grade—they may not learn the material fast enough or may struggle with required background knowledge—but will work to continue improving their understanding long after the class concludes, eventually far exceeding the instructor's original goals. This is the type of enthusiasm I hope to inspire in my students, both in classroom and research environments. I also believe teaching has a positive impact on the instructor's knowledge and his research. Not only does it help him understand the subject material better, but in my experience, the best student questions often have strong research implications. As such, I am always excited to teach and mentor students in computer systems—both at the introductory and graduate level—as doing so will improve their knowledge and re-enforce mine.

During my tenure at Carnegie Mellon, I have had the fortune to TA the master's level storage systems class twice, the second time as the lead TA. This class is close to my research area and, though its enrollment varies, can be composed of more than sixty students. In addition to weekly homework assignments, students are required to complete several real-world projects, such as writing FSCK and both designing and building an iSCSI RAID controller. These projects re-enforce the concepts taught in class and broaden the students' understanding by requiring them to make key design decisions that do not have definite answers.

Helping guide students toward good design decisions by hearing their rationales and pointing out contradicting or supporting examples—but never giving away what I would do!—was my favourite aspect of TA'ing storage systems. During these sessions, I would literally see the students brighten as they struggled with concepts and came to a new understanding of the material. The answer to many tough systems questions is "it depends"—there are no correct answers, only good justifications. As such, I believe it is extremely important for systems classes to incorporate such real-world projects to help students understand how to use the theoretical knowledge taught in class. Students learn by doing, making mistakes, and learning from them, not by passively listening to lectures or solving homework questions with definite answers. I hope to incorporate real-world projects into every class I teach, both at the undergraduate and graduate level.

I believe real-world projects are also the key to inspiring excitement and long-term interest, as they promote critical thinking and help students realize what they could accomplish with the knowledge offered to them. However, designing effective ones, especially for the introductory level, can be difficult because students often lack the background in computer science necessary to complete large meaningful projects. Simple projects, such as writing code to sort an address book, are just busy work and will not inspire students to continue in computer science. Instead, I believe introductory class projects should involve students filling in simple pieces of a larger project. Focusing such projects on "hot areas," such as mobile app development will further serve to build their interest. To build strong coding practices and prepare students for industry, I believe projects at all levels should also include student-led code reviews, group coding, and an emphasis on writing useful log messages.

Engendering long-term interest is useful, but it is also important to ensure students' understanding during lecture. For smaller classes, this can be achieved via socratic teaching, in which the instructor engages students in a dialogue that guides students toward self-discovery of key concepts and ideas. Also, incorporating problem solving sessions into lectures (or recitations) can help instructors obtain a sense of the students' progress and plan the next lecture. For larger classes, I am curious whether online tools, especially those that track an individual student's progress via high-resolution metrics, can be used to identify struggling students and automatically provide them with more resources in the specific areas in which they need help.
As part of my teaching responsibilities, I am also interested in exploring ways to improve the breadth of systems students’ knowledge. I believe one method for accomplishing this goal is to broaden projects so that they incorporate elements of user studies, design, and maybe even business. For example, instead of just writing code to create a terminal, students could also conduct simple user studies so as to explore the effectiveness of the various ways terminals can present feedback (e.g., showing text, playing a sound, or flashing the screen). I strongly believe that increased breadth of knowledge will strengthen the systems community, empowering them to more quickly create and explore new ideas. Broader projects may also encourage enrollment of those turned off by the hacker-ish aura often associated with computer science.

In addition to classroom learning, exposure to research is important for a good computer science education. At Carnegie Mellon, I have helped mentor several junior graduate students and one master’s student for his thesis. All of these students worked on projects related to my dissertation research. These experiences have taught me a great deal about advising students. For example, I learned quickly that students tend to be overambitious in their goals, so it is important to pare down their expected deliverables in a way that preserves their enthusiasm. My approach for doing so is to frame their contributions as part of a larger goal that they can continue to make progress toward after the project deadline. I have also learned that it is important to let students try their own ideas, even if I have reservations.

My experiences mentoring students have been incredibly positive. Most enjoyable has been watching them learn and grow as researchers. In no way is such progress better punctuated than in meetings in which a student will make an observation that I overlooked. I believe such mentoring opportunities are invaluable and plan to encourage my own graduate students to mentor junior graduate students and undergraduates.

In addition to my formal TAships and mentoring responsibilities, my recent experiences running a reading group, mentoring junior students on presentation skills, and volunteering to teach small children about science have confirmed my love of teaching and communicating information. I believe nothing will inspire a student as much as a committed instructor or mentor, passionate about his field, and always eager to help his students. I look forward to the continued opportunity to avail myself as one.

http://www.ece.cmu.edu/~rajas