

My goal as a teacher is to help students become knowledgeable independent thinkers and researchers. I try to encourage and challenge my students, foster critical thinking, and pass on the inspiration I have received from my own teachers. Most importantly, I try to create an environment in which students can realize that learning concepts and acquiring skills can be a natural side-effect of asking questions that interest them.

**Experience** My experience as an educator is diverse. It includes work as a teaching assistant and occasional lecturer in undergraduate operating systems at UT Austin, mentor-ship of junior graduate students doing systems research, and playing leadership roles as an engineer and systems designer in industry. It also includes over a decade of teaching one-on-one guitar lessons and group yoga classes. I bring a unique perspective to teaching computer science. I am often struck by how much the basic skills I use as a teacher generalize across such apparently disparate domains.

**Approach** My research style relies heavily on system building; consequently, my teaching style emphasizes cultivating the skills and qualities necessary to be a good builder of systems. I believe that successfully preparing students for careers in engineering or systems research, entails fostering four basic skills:

1. Strong implementation/troubleshooting skills
2. The ability to think creatively, abstractly, and analytically
3. The ability to observe with perspective
4. The ability to communicate effectively

My approach to teaching aims to cultivate those skills in my students, whether I am teaching in a classroom environment or mentoring one-on-one.

**Fundamentals: implementation and troubleshooting** Students must develop a strong command of fundamentals. This means understanding basic concepts and programming abstractions, as well as understanding how technologies interact to support those abstractions. Being able to write good programs and debug them effectively depend critically on these skills, and developing them requires practice. Because of this my teaching style emphasizes implementation as a means of evaluating ideas. I feel strongly that challenging programming assignments are a critical component of a good education in computer science.

I have invested a lot of energy in the development of and refinement of programming assignments for the undergraduate operating systems course at UT Austin. As a teaching assistant for Professor Emmett Witchel, I developed a programming assignment to teach the students about the MapReduce programming model. The assignment requires the students to develop a search engine using the Hadoop framework. That assignment is still a core assignment in Professor Witchel's classes. I also contribute to the ongoing refinement of the parallel programming assignment for that class. I have continued to contribute to that code base after my graduation when I have seen opportunities to improve its pedagogical value with changes and additional features.

Providing good implementation experience requires not just practice writing new code, but practice troubleshooting systems in different environments. Among the most rewarding aspects of being a TA for me came from investing time to help students improve their debugging skills, which required a willingness to work one-on-one with them. I found that even students who started the OS class as relatively experienced programmers were frustrated by the debugging process, and were unable to solve problems when an IDE-based debugger couldn't help them. By providing one on one assistance and suggesting simple techniques, assertions, I was able to move these students forward and allowed them to begin to generalize these techniques for problem solving independently. This kind of emphasis on implementation and trouble-shooting is central to my notion of what an education in computer science should provide.

**Analyzing problems and thinking abstractly** The centrality of abstraction in our field requires no reiteration. Helping students learn to think through problems and think abstractly about them is essential. To cultivate these skills, I try to guide my students by encouraging a collective dialog about how to solve a problem, starting with minimal assumptions and relying on first principles as much as possible. For example, when asked to give a lecture about MapReduce, rather than lecture about features and architectural details of the system, I simply asked the class "if you needed to search a body of text for keywords, how would you do it?" At the beginning, I specified no other aspects of the problem, and instead, pointed out fundamental assumptions the students were making in the solutions they proposed, or adding constraints to the problem as the students collectively converged on solutions.

The process of considering design decisions collectively to arrive at an understanding of why MapReduce is a useful framework is beneficial in two ways. First, it gives students practice reasoning through problems, and understanding the relationships between abstractions and implementations. Moreover, the process teaches students to trust their creative instincts and their ability to arrive at sound conclusions.

**Observing with perspective** The ability to recognize high level patterns is important for students who move into careers as engineers, and is essential for students who move on to become researchers. To be successful researchers, students must learn to see the similarities in disparate systems: for example Zyzyvva and transactional memory both represent systems that derive their fundamental benefits from speculation, despite the fact that these systems apply to different layers of the technology stack. Fostering the ability to recognize these patterns augments the repertoire of tools that students can rely on for problem solving, and is an important tool for making good decisions in selecting research problems. To foster this ability I rely on small-group format discussion and reading papers. As a teacher my role is to provide perspective, and encourage students to voice their observations and make connections with other systems and design patterns.

**Communication** Being able to communicate effectively is fundamental: the field cannot move forward without the sharing and collective evaluation of ideas. I emphasize both written and verbal communication in my teaching. I encourage students to refine their communication skills in many ways. I encourage them to work together on problems, which requires them to explain their ideas in a way that others can understand. As a teacher of graduate students I will emphasize oral presentation of papers as a core classroom activity, as well as conference-style write-ups for class projects. As a mentor of young researchers, I will rely on weekly one-on-one and group meetings to encourage open debate and collective, creative problem-solving will be the centerpiece of my approach to teaching and building a research group.

However, being a good educator requires more than just asking good questions; it requires creating a supportive, safe environment for exploration. Good communication requires practice, and practice inevitably entails making mistakes. Creating an environment where students feel safe practicing, improvising, making mistakes, and contributing their observations and insights in an authentic way is one of the most important things I can do as a teacher. In this area, I lead by example: I share my own love of the subject matter and provide perspectives that are often iconoclastic. I take risks, share my own thought processes, and encourage students to challenge me.

**Philosophy** Teaching is a wonderful opportunity to educate and influence that comes with an awesome responsibility. I enjoy teaching because contributing to students' intellectual growth is profoundly satisfying. Interacting with students and sharing ideas is symbiotic: as a teacher, the process of improving on how I communicate concepts helps me refine my own perspectives and command of the material. However, teaching is also a responsibility: passing on ideas and a love of inquiry is essential to a vibrant research community. Moreover, I feel a responsibility to students to educate them as well as I possibly can: staying true to that responsibility requires an ongoing commitment to improving my teaching skills.

I enjoy teaching very much. Good teaching requires a very careful balance of sensitivity, humor, acceptance, and willingness to demand the best from students while encouraging them and allowing them space to experiment. My most effective tools as a teacher are my own love of the things I teach and my willingness to devote time and energy to helping students, understanding their perspectives, and modeling my own approach to being a researcher. My love of the subject and my commitment to helping students understand the material is evinced by the strong positive response I received from the students in UT's undergraduate operating systems courses. An excerpt from one student's written assessment reads "Chris has gone above and beyond in showing his knowledge of this course and has been extremely helpful in debugging and helping us to understand the material and labs. This is a very difficult class and quality TAs are crucial. I'm so thankful that he has been my TA and probably doesn't get enough credit for the amount of time he has spent helping create new and updated labs, spending time with students, and catching even the Professor's mistakes. Seriously, you rock!"