As a shy person, it took me some time to recognize how deeply satisfying I find teaching. But when I noticed how much I enjoyed giving science-oriented weather observatory tours, serving as an introductory computer science tutor, and giving research presentations, I started to recognize that teaching is the career for me.

Ever since this revelation came late in my undergraduate years, I have single-mindedly pursued the goal of becoming a faculty member at a teaching-focused institution. After experiencing the value of an undergraduate education at such an institution, I have explored various other settings: I worked as a professional engineer, did research in industry, and did my graduate work at a large research institution. Throughout all of these experiences, I have only become more certain that my calling is to teach at an institution where innovative and high-quality teaching is a priority.

During my graduate program I have sought out numerous opportunities to gain experience, hone my teaching skills, and deepen my understanding of student learning. In my approach to teaching, I focus on two guiding principles that I believe set the stage for quality learning:

1. Students need to interact with the material, and they should do so as early as possible.
2. The instructor should strive to understand the diverse perspectives of different students.

I will begin by discussing these principles and how I have put them into practice, then show an example of another way I'd like to approach these ideals by teaching good question-asking skills.

Interaction

When students interact with the material, they learn better and I teach better. Students learn better because they become involved in their own learning; I teach better because I get insight into their learning process.

Giving excellent lectures is difficult precisely because students aren't forced to engage with the content. As an instructor for a large lecture course on data structures, I knew how easy it would be for students to zone out, or to see the material but not really ingest it. Although the class size precluded the use of most active learning techniques, I started giving a handful of 1-2 minute online polls during lecture. The polls were most effective when many students got the questions wrong; students got an early wake-up call that they didn't understand the material, and I gained insight into the common misconceptions that needed to be addressed. A tight feedback loop helps students directly and indirectly by allowing me to make adjustments quickly. This simple tweak to an existing lecture course cost only a few minutes of class time and had tangible benefits: many student evaluations specifically praised the use of polls and asked for more of them.

Courses with fewer students and/or less rigid curriculum present more exciting opportunities for interactive learning. When teaching small lab sections of an honors introductory MATLAB programming course, I designed programming activities that used concepts like turtle graphics, image convolution, and Conway's Game of Life to reinforce and apply the basic programming concepts from lecture. The self-guided format forced all students to be engaged with the material, while remaining adaptable to the students' ability levels. This was one of the most satisfying teaching experiences I've had, because I was able to simultaneously devote personalized instruction to the students who struggled the most while keeping the strongest students engaged by including extra challenges in the exercises. I aspire to bring this level of interactivity and adaptiveness to all courses I teach; this is why I want a job at a small school that values innovative and high-quality teaching.

Understanding Diverse Student Perspectives

Learning is a challenging and messy process: not all strategies work for all students. A tight feedback loop supplies feedback, but in order to react effectively to this feedback - and in fact to teach effectively at all - I believe it's critical to understand the student's perspective.

The most direct consequence of this principle is that mistakes matter. Mistakes provide some of the best learning opportunities, as I saw when the polling questions that students got wrong did the most good. Although mistakes take more time to understand than to correct, student and instructor both benefit from analyzing how the mistake came to be. The empathy and acknowledgement that come with taking time to understand the mistake also helps students feel like the instructor is invested in their learning.

Understanding student perspectives is especially critical for teaching diverse groups of students. Issues like stereotype threat, impostor syndrome, and extracurricular skills gaps, can subtly discourage students - especially those from diverse backgrounds - from persevering in computer science. I consider this problematic first because I want all of my students to succeed, and it is unfair that some of them face these obstacles. Secondly, I believe that the world stands to benefit from people with new and diverse perspectives applying the tools of computer science to problems that affect their lives. It is naturally easier for me to understand and empathize with people whose backgrounds are similar to mine; for this reason, I feel strongly that understanding the lens through which students from underrepresented groups experience my teaching is critical.

I will touch on three strategies to combat these forces that hinder diversity. First, getting feedback: listening to students is one of the best tools I have. Second, I take outreach seriously: I have participated in many programs that expose underrepresented student populations to STEM (see my CV for a comprehensive list). Understanding the perspectives of young students encountering computer science for the first time has been particularly valuable. One strategy I use is to appeal to a wide range of interests by allowing students to explore the creative possibilities enabled by programming. For example, when supervising self-guided projects for high-school girls, one group got much more interested in programming when I guided them towards using MATLAB to implement creative Photoshop-like image filters.

Good Questions

The ability to be a self-sufficient and inquisitive learner is perhaps the most important take-away from an undergraduate education in any field of study. My experience suggests that these skills can be developed by getting students in the habit of asking detailed, well-researched questions. The process of due diligence before asking a question helps students learn to specify their question carefully and find answers themselves where possible; clearly formulating a question requires organizing and communicating their thinking in the presence of uncertainty.

One technique I learned in my training as a telemark skiing instructor is to have students try to perform a task before being told or shown how. In a classroom setting, this principle can be applied by asking students to solve a problem, or formulate a question whose answer would help them solve it. This exercises their creativity and problem-solving skills, improves their comfort in unfamiliar situations, and increases their investment in the instruction that follows because it relates back to their experiences.

Getting students asking good questions has many benefits that tie together the principles outlined above. First, asking questions serves as a means of interacting with material. In one particularly excellent course I took as a student, we were required to send in questions about the readings; this caused me to read the papers more critically and understand them much more thoroughly.

Secondly, as an instructor, student questions give me feedback and insight into student perspectives. When good questions that identify specific gaps in knowledge or understanding are commonplace, it creates a classroom culture in which the purpose of class is not knowing things, but learning things. Many students are afraid to admit that they don't know things. Everybody benefits when such admissions are constructive and commonplace, but especially students who face obstacles like impostor syndrome and stereotype threat.

Conclusion

My love and enthusiasm for teaching computer science and my respect for the challenge make me ideally suited for a career teaching computer science. I am constantly seeking ways to improve: I am currently taking a course on Teaching in Higher Education to improve my understanding of pedagogy. Although I have received excellent student evaluations and high praise from co-instructors, effectively teaching diverse groups of students in finite time remains a boundless challenge. To me this challenge is the most exciting one, and I look forward to a career spent improving my teaching, serving students better and helping them succeed.