TEACHING STATEMENT

ANDREW QUINN

My passion for computer science, research, and engineering traces back to individual mentors and teachers throughout high school, college and graduate school. With the exception of my wife and immediate family, my most important relationships have been with these mentors and teachers. My experiences mentoring early-career graduate students have been equally rewarding. A key reason why I am pursuing an academic job is to pass on my passion for computer science to undergraduate and graduate students through the close relationships that develop from teaching and mentorship.

I find that success in computer science is more often determined by early exposure to computer science topics than aptitude. Therefore, it is my main goal as an educator to foster an equitable learning environment in which all students can succeed regardless of their backgrounds. Based on my experiences as a walk-in tutor during my undergraduate at Denison University and as a graduate student instructor at the University of Michigan, I find that encouraging active engagement from students, modeling my own thought processes, and providing many alternative explanations of material help foster an equitable learning environment.

In my teaching experiences, I observed that many students build an understanding of computer science concepts, but struggle to apply their knowledge in new contexts and domains. To help students be successful outside of their coursework, I will build pragmatic assignments and course projects that help students gain experience and skills in applying their knowledge. In the future, I plan to design entirely new courses to help students better hone these skills.

TEACHING AND MENTORSHIP EXPERIENCE

TEACHING EXPERIENCE. I was a graduate student instructor for a data structure and algorithms course with over 100 undergraduate students enrolled in the spring of 2019. My duties included creating and presenting material at two two-hour lab sessions each week, holding drop-in office hours, and writing and grading exams. I was a walk-in tutor as an junior and senior at Denison University; due to the small size of the computer science department, these sessions often acted as office hours for students in introductory computer science courses.

TEACHING PHILOSOPHY. When teaching, I aim to foster an equitable learning environment. During direct instruction, I speak excitedly about the material to show how passionate I am about computer science. I provide multiple explanations for the same material, with the aim that students who learn differently may find one or another approach more understandable. I provide analogies (e.g., relating sorting algorithms to how students are organized by height for picture day), but do not rely on them, lest my lessons be inaccessible for students from different backgrounds.

I structure my material to emphasize active learning, which fosters equity since all students gain a structured environment for hands-on learning. I begin sessions with simple "do now" exercises that apply concepts from earlier in the semester to solve new problems. These activities not only encourage engagement, but also help students better apply course material in new scenarios. In addition to coding exercises, I facilitate active participation by frequently using "think-pair-share" activities and structuring the course material in a question-answer format.

In addition to promoting active engagement, a question-answer format provides ample opportunities for me to model my own thought processes for students. When modeling my thought processes, I talk through many possible solutions to each problem and explain their pros and cons. I explain my thoughts naturally, even when this highlights my mistakes! By highlighting the times when I am wrong, students gain an appreciation for the challenges of computer science and see that even experts make mistakes. I believe that these subtle nudges can make a big impact in promoting a equitable environment.

MENTORSHIP. As a senior graduate student, I have had the pleasure of mentoring a number of early-career graduate students. I have seen how each graduate student has their own strengths and weaknesses; as mentors we must identify gaps in student's knowledge to help them grow. I am not a passive mentor. Instead, like in teaching, I mentor by modeling. I explain my thought process in depth, regardless of whether I am

discussing research problems, how to best organize a paper, or construct a conference presentation. I like to ask extremely basic questions and talk through a very broad set of possible solutions. I am transparent in my mistakes and I am frequently wrong. By explaining everything, not just my best ideas, I find that students can better understand how to evaluate ideas and are less nervous to bring their own ideas forward.

I have found modeling to be an effective technique for many of the students that I mentor. In particular, I am currently mentoring Ian, an early-career graduate student, on projects that improve the reliability of persistent memory applications. Ian has fantastic technical ability and a great sense for how his research fits into the broader systems community, but struggles with technical writing. When writing Ian's first paper, I spent extra time with him walking through many possible ways we could organize the paper. I showed him how to simplify his writing and shy away from a narrative voice. When we submitted follow-up work, Ian was able to organize the paper on his own and was better at limiting narrative statements.

FUTURE TEACHING PLANS

In the future, I aim to foster an equitable learning environment by using the techniques that I found most useful in my past experiences, encouraging active engagement from students, modeling my own thought processes and providing many alternative explanations of material. I would be excited to teach operating systems, distributed systems, and software engineering courses at the graduate and undergraduate levels.

I find that many students struggle to apply concepts from coursework to new domains, and I would like to design courses that make this a primary learning objective. I envision an upper-level undergraduate course centered around a large core project. Students will work in groups to build a large system, such as a recommendation system, social network, or search algorithm. The course material will explore recent trends in systems research that may be applicable to student projects, such as systems for ML, heterogeneous systems, kernel bypass models, etc. By identifying how recent research can be adapted to their projects, students will work on applying computer science concepts to new domains. In addition, I imagine that each group will be fairly large (8–10 students), so that students can gain experience working on a large code base, where they do not understand how every module works. Moreover, a larger group allows students to work on important soft skills needed to be successful engineers, including collaboration, technical writing, and giving presentations.

CONCLUSION. A main reason why I am pursing an academic job is because of the awesome impact that a good mentor and teacher can have on a student's life. My primary goal when teaching is to create an equitable environment by using inclusive teaching methods. As a mentor, I have found that transparently modeling my thought processes offers a fantastic mechanism for graduate students to learn by example.