

TEACHING STATEMENT

AMY X. ZHANG

Much of my formative education that defines who I am today took place on the tennis court in addition to the classroom. As a competitive tennis player, my daily practices oscillated from repetitive drills tweaking a stroke to practice matches where I tried out different styles of gameplay with teammates. Together, these practices allowed me to adapt my strategy on the fly when it came time to compete on my own. This *process-focused* training that I took part in as a tennis player and later taught as a tennis coach also has parallels in academia. When it comes to teaching and mentorship, I focus on process by helping students develop a solid foundation and by guiding them through open-ended exploration towards conducting independent work. The following three main pedagogical principles summarize my approach to teaching and mentorship.

First, I create opportunities for students to *develop their own initiative*. In classrooms, I implement flipped classrooms, facilitate open-ended discussions, and assign projects where students have the flexibility to decide what they are creating. While mentoring students on projects, I press students to develop their own research tastes by learning what ideas excite them and giving them the room to pursue them.

Second, I *emphasize skill-building* when working with a student so that they can build a foundation for working independently. This often means starting by focusing on understanding the underlying technology, relevant libraries, and code. When mentoring other research students, I emphasize developing writing and presentation skills by encouraging them to write early drafts and submit posters and works-in-progress.

Finally, I *meet students where they are* with regards to their daily lives, passions, and diverse backgrounds and abilities. In the user interface design course that I taught, I noticed that students responded much more thoughtfully to designs from their everyday lives. As a result, I designed examples during lecture grounded in students' experiences with technology when explaining design concepts. As a research mentor, this means tailoring my mentorship style to fit the needs of the mentee, such as varying the level of structure or emphasizing development of research versus programming skills.

TEACHING EXPERIENCE

My very first experience with teaching occurred in my junior year of college, when I taught a freshman seminar to 18 students exploring topics in computer science at Rutgers University. I developed my own syllabus, discussed different careers, held mini-hackathons, and organized a field trip to Google. It was particularly exciting to be there at the start of my students' journey with computer science, to ignite that spark and assist them through their undergraduate career, including recommending them to jobs and research labs.

My most defining experience was when I was an instructor with Professor David Karger and Lea Verou of MIT's User Interface Design course in Spring 2018. I had been a teaching assistant for the class the year before and had taken the class as a student in 2014, but leading the class as an instructor was a unique challenge. Nearly 250 students enrolled in the course, necessitating the hiring and management of 9 TAs. The class was taught as a flipped classroom, where reading assignments were done as homework and class time was dedicated to in-person discussions and collaborative exercises. While much of the materials existed from prior iterations of the course, I significantly updated all of my lectures and in-class exercises to draw from more up-to-date examples. For instance, in one lecture on safety, I led an exercise on "accidental likes" on social media and whether designs should try to prevent errors or allow more graceful undo capabilities. In a different lecture, I led a discussion on the ethical and user experience issues behind chatbots like Facebook's M, where the system may be deceptive about whether users are interfacing with an A.I. or a human. One student said in their course evaluation, "*Amy's lectures really contributed to my knowledge of important aspects of UI design. I really liked how Amy led in-class discussions and was able to explain UI issues and solutions with conciseness.*"

Since the class was taught as a flipped classroom, we assigned readings before each class. In a case of "dogfooding" my own research, we used the Nota Bene (NB) tool, a research tool developed in my lab, to allow students to have annotated discussions "in the margins" of the text. This promoted deeper engagement with the reading material as well as social learning.

In addition to lectures twice a week, we also offered three versions of the course simultaneously, to serve the different needs of undergraduates, masters students, and Ph.D. students. This meant juggling three different sets of assignments, as well as creating grading rubrics, and coordinating and holding office hours for all three. Of office hours, one student said *“Very patient and laid back. Great lecturer. Really helpful during office hours and didn’t get overwhelmed by enormous amounts of people”*. I also organized a separate research seminar for the Ph.D. students where I lectured on how to gather empirical data and design user study experiments and students read seminal papers in human-computer interaction research.

Finally, the class had a group project component where groups met in smaller studios led by TAs once a week and took a project from ideation, to paper prototype, to computer prototype, with several rounds of user testing. With over 50 groups, this was a massive coordination effort. At the end of the class, we held presentations and gave out awards for the different group projects. For instance, one group that won an award was able to develop and test a working prototype of a user interface for autonomous wheelchairs in hospitals.

Despite teaching such a large class, I was still able to get to know many students one-on-one through office hours and after class. In one case, I went to a sushi buffet lunch with several students through a program at MIT that allows students to invite any of their instructors out for a free meal. In another case, I gave advice to a student in my class on the user interface component of her masters research project. One student remarked, *“I personally liked this instructor the most. She seemed to care about us a lot.”*

MENTORSHIP EXPERIENCE

When it comes to mentorship, I am proudest when I see my students graduate from following my lead to taking initiative and charting their own direction. Over the last 5 years, I have mentored 14 undergraduate students on both research and engineering projects over the course of a semester or year, as well as 3 masters students on their masters thesis projects. I have also successfully mentored students from traditionally underrepresented groups, including 10 women.

One of my mentees, Kaitlin Mahar, worked as a undergraduate with me for a year doing software development before pursuing a research project for her M.Eng. thesis on designing a tool for people experiencing online harassment. I guided her through difficult interviews with people facing harassment, design and development of the tool, a user study with 10 participants, and the writing of a research paper published at CHI 2018, which she presented. Beyond research, she enrolled in the Mozilla Open Leaders Program to make Squadbox accessible for open source contributors and also helped lead a workshop on combating online harassment at Mozilla Festival.

Another one of my mentees, Jane Im, joined our research group from University of Korea while an exchange student at MIT in her junior year. Her work involved collecting the first comprehensive dataset of deliberations on Wikipedia, spanning 7 years, as well as conducting qualitative and quantitative analysis, and developing a model to predict whether a deliberation would eventually get resolved. This work was published at CSCW 2018, where she was the presenter and lead author. She also presented this work during the Wikimedia Foundation Research Showcase. She is now a Ph.D. student at the University of Michigan in social computing.

COMMUNITY

Mentorship and peer advising need not be confined to formal research relationships within my research group. I have also been a part of many communities during my Ph.D. that have enriched my work and grown my network of interdisciplinary collaborators, including not only other academics but also journalists, activists, and civic groups. I have been involved with expanding the HCI community in my local area, including leading the Boston CHI Labs group, which puts on social events for local HCI researchers, as co-president for the last year and co-organizing the MIT HCI Seminar Series in Spring 2017.

I have also supported undergraduate development outside of research and the classroom. For the first four years of my Ph.D., I served as an undergraduate resident advisor at Harvard University in the Cabot House undergraduate dorm, where I organized social events around the house, study sessions related to computer science, and helped students with their career development.

FUTURE COURSES

In this section are courses that I would look forward to teaching.

Undergraduate Courses:

Human-Computer Interaction: A practical introduction to the building blocks of human-computer interaction, including the steps of needfinding, ideation, iterative prototyping and critique, and evaluation. Students would be taught about fundamental concerns in design, as well as well-known heuristics. The class would involve a semester-long project, studio feedback, and teamwork.

Programming for Data Science: An introduction to programming languages, statistical models, and tools necessary to conduct data analysis. Students will learn the basic data pipeline, from acquisition and cleaning, to data storage and database querying, to analysis and machine learning, to visualization and presentation.

Introduction to Web Development: A primer on how to develop and deploy web applications, including client-server architecture, database design and hosting, and popular web development frameworks such as Django. Students will learn front-end languages such as HTML, CSS, and Javascript.

Graduate Courses:

Social Computing Systems: Exploration into the design of computer-mediated social systems. Topics include moderation strategies, anonymity, motivations, newcomer support, and interface and algorithm design. Students will prototype new social computing systems, study existing ones, and learn research topics in this field.

Computational Social Science: An introduction to quantitative methods and tools for data analysis of large-scale social data, including data collection, network analysis, text analysis, and machine learning, coupled with readings and discussions around relevant social science theories, research ethics, and practical applications.

Crowd Computing: An introduction to techniques, workflows, motivations, and algorithms for crowdsourcing and human computation. Exploration of applications including in paid crowdsourcing platforms, peer production, the future of work, and collective intelligence such as citizen science and participatory democracy.