# TEACHING PHILOSOPHY Iris Howley

I am driven by a goal of inclusivity in the classroom which is reflected in my project-based learning and evidence-based teaching philosophy. My approach focuses on alignment of learning goals and assessment to construct learner-centric support. To further improve the learning experience, I gather early course evaluations from students and seek assessments of my teaching from pedagogical experts. Through leading two recitation sections at the college-level, mentoring junior students in computer science, and supervising independent study projects, I have formed the following teaching philosophy.

*Project-based learning for inclusivity.* Project-based courses with multiple deliverables distributed throughout the semester are especially conducive to increasing inclusivity in computer science. As a two-time member of the organizing committee for the 3-day Opportunities for Undergraduate Research in Computer Science conference (OurCS) for young women, I could see firsthand how contributing to a real project lead to shifts in the young women's self-perceptions and relationship to computer science research. Many participants stated a renewed interest in pursuing computer science research, which is a sentiment I shared when I attended OurCS as an undergraduate years prior to attending graduate school. Similar feelings were shared by participants in a mentoring program I co-taught focusing on using the Lilypad Arduino. The middle school girls were part of a foster care program, and we presented the textile computing curriculum as a form of art therapy. While the light-up teddy bears and sweatshirts were certainly works of art, it was the girls' newfound confidence in working with technology and design that truly stole the spotlight. Not only could they program behind a computer, but they could proudly wear their creations in their schools and communities.

Furthermore, computer science courses often lend themselves well to project-based assignments as a first step toward real world learning. Both the recitation sections I taught as a teaching assistant contributed to a semester-long project, and I observed that students would return to previously submitted sections in order to improve their final project. This motivation to expend extra hours revisiting prior concepts is a nice additional side effect of project-based courses and assessments. My approach to teaching introductory human-computer interaction and user-centered design courses involves several projects that build upon each other, involving heuristic evaluations, competitive analyses, construction of cultural models, think aloud methods, and prototypes at a variety of fidelities to result in an iterative final project.

*Learner-centric support.* Research shows that students learn better when information is provided to them in a variety of formats. I share my slides with students before lecturing, and they include a mixture of text and visuals. I often include group work activities, which helps students practice skills and apply memorized knowledge. When instructing students on procedural skills, providing practice opportunities and feedback on that practice in a recitation or lab is useful so student misinterpretations are not carried into the homework or other summative assessments. This practice of skills and emphasis on technical communication are important aspects to joining the professional community of computer science.

Students have varying strengths that require an individualized support plan. I generally hold common weekly office hours, but gladly add alternative office hours at students' requests. In some cases, students have family or religious constraints that require extra consideration, but sometimes a student just needs extra time or practice with a particular concept. When I taught a human-computer interaction prototyping class to a mixed class of undergraduates and graduate students from psychology and design backgrounds, completion of a programming class was a prerequisite. However, the first homework assignment revealed that one student did not remember how to debug a programming assignment. In this case, I initiated an extra office hour with this student to focus on the general principles of debugging.

*Explicit about what students will learn*. My lecture slides as well as my syllabi start with a "After this course/lecture you will be able to…" bullet point list. It is important to be explicit in learning goals, not just so students know what they should be learning, but also so it is easier to ensure as an instructor that those goals are covered, achieved, and assessed. These explicit expectations focus on observable verbs,

such as “perform a structured interview” or “list the differences between supervised and unsupervised machine learning methods”, and not words that are harder to actually observe, like “understand.”

Most courses also have implicit skills that are expected to be learned or performed. Often these implicit skills are not core domain knowledge, but without an ability to collaborate with peers, read critically, or communicate effectively, the domain knowledge is considerably less useful. In my classroom, these implicit skills are included explicitly in the syllabus, daily goals, classroom instruction, and grading guidelines. As the classes I teach are usually project-based, the communication expectations are particularly important to impart.

*Assessment as a diagnostic tool for improving the learning experience*. When teaching interdisciplinary courses to mixtures of undergraduate and graduate students, I learned quickly that a quick initial assessment of student programming background is useful for tailoring the lecture and assignments to student capabilities. However, these early assessments do not have to be quizzes, and in one case a very simple homework assignment was quite revealing about the mental models of my students. This past year I guest lectured three seminars for a survey course for learning science doctoral students. My lectures included an introduction to machine learning in education and text mining. When students submitted sample data sets to text mine during the third hands-on class session, students who provided small data sources, such as a single essay, quickly revealed where more time and explanation should be devoted in my text mining process lectures. In this way, I approach assessment as part of a larger mentoring approach.

When I taught the User-Centered Research and Evaluation recitation, I invited a pedagogical expert to observe one of my classes and provide feedback. I also elicited formative early course evaluations from my students. When I collect this feedback, I present a brief overview and response to my students so that they know that I am listening. From this particular course, students liked how I summarized the lecture at the beginning of the recitation, and they also appreciated the small group work and feedback provided at that time. Suggested improvements mostly centered around providing more details about the homework assignments, which was logistically difficult due to the timing of the class’ team teaching. While the feedback is not easily summarized numerically as it is open-ended, I have made the external review and these course evaluations publicly available online: www.irishowley.com/website/pUCRE.html

*Evidence-based teaching.* As an interdisciplinary human-computer interaction researcher and learning scientist, I have continuously used education research to inform my pedagogical approach. I completed courses in science-based research in education, and a pedagogical course on aligning educational goals, instruction, and assessment. Part of my research often requires developing and delivering instructional content for students in the primary, secondary, and undergraduate levels. I also taught mixed classes of undergraduate and graduate students in the school of computer science as a recitation instructor.

My prior teaching experience prepared me to instruct at the undergraduate and graduate levels in computer and information science. In particular, I am prepared to teach introductory human-computer interaction courses, more advanced prototyping or interface design courses, specialty courses on educational technology and educational games, as well as a range of introductory and intermediate computer science courses. I enjoy opportunities in which to incorporate online discussions and hands-on in-class project work into my teaching.

Teaching and education are an integral part of my research career, as my research most frequently focuses on educational contexts. When I read prior literature, investigate questions on computer-supported collaborative learning, study sociolinguistics in the classroom, or research student help seeking behaviors, all of that time spent also works toward improving my teaching. As a computer scientist, the learning sciences were the most natural progression to develop my interest in teaching and now as a scientist and instructor it is great to have my teaching inform my research and vice versa. I especially appreciate how the result of my research and teaching efforts can both contribute to the same effort: developing student skills so they can be effective participants in their professional communities as well as the community at large.