Courses

One of my strengths as a teacher is my diverse computing background: I've led software-engineering teams for desktop-, and web-, and mobile-app development; I've explored academic-research topics in computer architecture, database systems, and artificial intelligence; and, as a researcher for the Walt Disney Company, I've collaborated on such innovative projects as robotic display swarms for theme-park attractions. These experiences expand the set of topics about which I can instruct, as well as enrich the content I present.

I am fully qualified to teach introductory computer science (CS) topics (e.g. programming, algorithms/data structures, discrete math/theory, computer organization, etc.) and also bring strong expertise of more advanced topics in the areas of databases and artificial intelligence. I also have software-development experience to bring to bear on such practical courses as web-app development and capstone project courses.

I see developing curricula as an opportunity to deeply understand a field, both for research and personal development. One course I would propose, Cognitive Systems, would focus on efforts to develop computational agents that exhibit human-level intelligence: an examination of such historical artifacts as ELIZA and the General Problem Solver (GPS); modern research paradigms, including cognitive architecture and psychological modeling; as well as breakthroughs in, and limitations of, such systems as Watson and Siri. The second course, Computing for the Greater Good, would focus on a socially relevant theme each term, such as privacy & security, open governance, or technology in developing regions, and would integrate readings of current events, computing research, and relevant tools/techniques, culminating in a final team project. For example, an apposite exploration would involve news stories of the Arab Spring, Stuxnet, Anonymous, and Edward Snowden, contextualized with modern security research and cryptographic tools, with possible project topics relating to the privacy issues of electronic medical records, sensor/communication systems, and social networks. Over the term, student teams identify a relevant problem in this space, as well as develop and evaluate an appropriate software solution; project outputs would ideally lead to published research findings and/or serve as the basis for entrepreneurial startups.

Experience

At North Carolina State University, I served as the teaching assistant (TA) for the graduate Database Systems course, where my responsibilities included holding office hours, grading assignments, developing/ proctoring/grading exams, and also taking primary responsibility for the semester-long team project. This project involved four-person teams interpreting a requirement specification and producing both an Oracle database schema and a Java-based user interface. I developed the project specification, regularly consulted with student teams on progress, and graded mid-term and final project outputs.

I engaged in a variety of teaching-related roles at the University of Michigan. First, I served as TA twice (Intro. Data Structures & Intro. Computational Theory). In both courses I contributed to a teaching team involved in the development and grading of assignments and exams. I ran office hours and weekly discussion sessions, for which I developed lecture and interactive problem-solving materials. Second, I was the Supplemental Instruction Leader for a college-wide introductory programming course (MATLAB and C++), which involved developing and delivering weekly 2-hour interactive sessions of supplemental lecture, question-answering, and problem-solving practice. Third, I produced and delivered more than a dozen tutorials related to the Soar cognitive architecture: these interactive sessions have been 1-8 hours in length and included student, faculty, and industry participants. Fourth, I served as a research mentor to multiple high-school, undergraduate, and graduate students. And finally, I led the graduate-student association in CS for two years, and both founded and led SIG-Faculty, a student group to prepare graduate students for careers in academia.

I have also engaged in a variety of teaching-related outreach opportunities. For example, for two years I developed curricula for, and instructed in, Michigan's LEAD Engineering, a residential summer program that brought under-represented K-12 students from around the US to take advanced engineering courses, develop leadership skills, and engage challenging projects from a variety of industry partners in 13 different departments. My other outreach venues have included consulting with Powering Potential, an NGO providing open-source, solar-powered classroom equipment and training in Tanzania, and Random Hacks of Kindness, where I served as a project mentor for students in Newtown, CT following the Sandy Hook massacre of 2012.

Philosophy

I am passionate about computer science and I strive to develop an inclusive learning environment where I can share my excitement for challenging ideas and problems, while being respectful of and responsive to student needs. I work towards these objectives in four ways:

I adapt to student needs. Students come to a classroom with a variety of needs that stem from diverse backgrounds/interests, distinct understanding of prior material, and differing learning styles. I continually adapt course material/presentation and classroom management in an attempt to create an optimal environment for each student and the class as a whole. To engage students, I enhance traditional lecturing with active learning techniques, such as Think-Pair-Share and role-play (e.g. physically acting out algorithms or data structures). I also strive to develop a supportive classroom environment by incorporating collaborative small-group activities and both encouraging and rewarding student involvement. To touch upon a variety of learning styles, I seek out and share a variety of resources pertinent to course topics, including online videos/talks/lessons, academic papers, as well as open-source projects. Additionally, I have found that many students benefit most from small group and/or one-on-one interaction during discussions/labs, as well as office hours. These personalized sessions are some of the most challenging and rewarding for me as an educator, and I strive to ensure that no student walks away without added comprehension of and appreciation for classroom material, as well as computer science as an intriguing and rewarding pursuit.

I set clear expectations and provide frequent feedback. Students are motivated when they understand the goals of a class, can visualize their path to achievement, and are provided clear sources of feedback along the way. I invest time and attention to clearly and consistently communicate my expectations involving learning objectives, assignment quality, and class participation. Setting a clear and challenging bar motivates students to focus their energy on mastering material, as opposed to juggling administrivia and/or guessing instructor priorities. I regularly begin sessions with an overview of upcoming examinations, highlights of potentially confusing assignment specifications, and a short-term contextual map of topics to come. In my experience, once we address these pragmatic sources of anxiety, the class as a whole can focus and engage the material of the day. I also strive to provide diverse, frequent sources of feedback, such as questions during lecture, one-on-one discussion during office hours/electronic communication, and fast, but informative, turnaround in assessment. Closing the feedback loop catches conceptual inaccuracies early and helps to make students feel supported in their learning journey.

I establish context and communicate my passion. Students that view course material as relevant and interesting are more likely to ask clarifying questions, think analytically about the implications of core concepts, and explore material beyond the confines of the course. I demonstrate relevancy by connecting course material to current events and industrial applications. For example, I have made contact with social networking, IBM's Watson, and environmental conservancy in data centers, such as to motivate the challenges of efficient searching and sorting. It has also been my experience that passion for a subject is infectious, and that students, in part, engage course material because a teacher exhibits excitement and interest. Thus I often share relevant problems and issues from my own experience, which includes academic/industrial research and entrepreneurial pursuits. I find that engaging in discussion of real-world problems, especially those in which I am personally invested, reinforces course material and encourages students to think about how they will apply computer science to improve the world around them.

I encourage a balance of exploration and analytical thinking. One of the distinctive characteristics of computer science, as compared to other engineering disciplines, is the relatively low level of resources and expertise needed to experiment with new technologies, contribute to real-world projects, and learn from practitioners. I actively encourage students to prototype ideas, contribute to open-source projects, and seek guidance from peers and relevant experts. However, analytical thinking is crucial to success, both in computer science and other endeavors, and so I seek to balance exploration with analysis. For instance, I encourage students to reason through processes and hypothesize outcomes before beginning an assignment or debugging a project. It has been my experience that exploration facilitates learning breadth and speed, while analytical thinking promotes deep understanding and engagement.