Teaching Statement

John Vilk
jvilk@cs.umass.edu

My primary goals as a teacher are to help students understand, retain, and apply course concepts. Toward these ends, the key aspects of my teaching philosophy involve engaging students using a conversational style to clearly motivate and explain concepts, and designing project-based assignments that require students to constructively apply class concepts to achieve a concrete goal. I applied these techniques while completely redesigning and co-teaching the popular undergraduate web programming course at the University of Massachusetts. I have also provided research and professional mentoring to several undergraduate and graduate students.

Teaching Experience

I completely redesigned the curriculum and assignments of the web programming course at UMass (COMPSCI 326) to provide students with the knowledge and experience required to write modern web applications. I had an unusually central role in shaping this class because my research focuses on web development, which uniquely qualified me to teach the material. The new material covers a wide variety of important topics including client- and server-side JavaScript, RESTful services, entity-relation (ER) diagrams for modeling data, and secure authentication mechanisms.

Since no standard textbook exists that covers all of these topics, I wrote a series of detailed workshop assignments that explain each concept while leading students through the process of developing a Facebook clone. These workshop assignments encompass the entire curriculum of the class and form the basis of most lectures that the instructor prepared. I care about open access, and designed these workshop assignments to be usable outside of the course and placed them publicly online.[1] Other UMass students have used the workshops to teach themselves web development, including one student who secured an internship at Facebook. In addition to planning the course and preparing all assignments, I also gave a lecture and engaged with the class extensively via online forums, detailed assignment feedback, and well-attended office hours.

Teaching via Conversation

I believe that students are most motivated to learn when the course materials (1) are appropriately designed for their level of understanding, (2) clearly illustrate why specific concepts are important and/or useful, and (3) are framed in the context of a compelling narrative or problem. Students are then able to comprehend the concepts presented in class and how to apply them.

Every workshop I wrote for the web programming class uses a plain English conversation with the student to naturally introduce concepts while working through solutions to specific, concrete problems. For example, the workshop on authentication covers hash functions, salts, encryption (HTTPS), and session tokens while discussing how to add a secure authentication mechanism to the client and server of a Facebook clone. Students were motivated to work through these workshops because they described how to *build Facebook* – a concept that visibly excited the class on the first day. Since the class had few prerequisites, the workshops covered a wide variety of important topics needed to help students succeed in the course including terminal usage, version control, debugging, and data modeling. Along the way, workshops linked to websites with deeper coverage of topics that the class did not have time to explore in order to stimulate motivated students that wanted to learn more; one student specifically commented that these workshops "inspired me to pursue learning web technologies on my own outside of class."

Project-based Learning

Projects solidify a student's understanding of a concept and help them retain knowledge by rooting it in a specific problem they had to overcome. I believe every course can benefit from the inclusion of project assignments that combine theory with practice. I designed the web programming course around a single

[1] An archived version of the material can be found here: `https://jvilk.com/cmpsci-326`

group project that motivates each unit of the course. Every student belongs to a "startup", and each startup chooses a "product" to build over the course of the semester. Each startup needs to complete a set of milestones during the semester, and each milestone dovetails with a set of workshops and lectures that introduce the concepts needed to complete it. When students became stuck on a problem, a pointer back to the relevant course material typically addressed their concerns and helped them understand the material in a new light. One student says that their startup experience resulted in an "incredible project that I have been able to speak about in job interviews and... motivate[d] me to continue my studies with a focus on software development and web technologies."

For a group project to succeed, group selection needs to be fair. I believe instructors should take an active role in group selection to ensure that individual students are not unfairly disadvantaged. For example, the web programming class seats 100 students, has students with varying levels of core CS topic expertise, and comprises about 15% students from neighboring colleges.[2] Randomly creating groups would leave some groups with gaps in their knowledge and mix students that lived far away from each other.

When forming groups I took into account specific information about each student. I developed a survey that asked students to describe their experience with relevant computer science topics and used this information to form groups with balanced expertise. To remove barriers to in-person collaboration, I grouped non-UMass students with peers from their school. I also took into consideration each student's group preferences. The instructor and I advised the students that they could talk with us about group grievances, which a few students did to warn us about non-contributing members. Overall, the groups worked well; students chose to sit with their group during class, arranged regular meetings, and worked together with group members to understand class concepts.

Mentoring Experience

In my role as a graduate student in the PLASMA Lab at UMass, I have mentored a junior graduate student in the lab, three undergraduate students on research projects through Google Summer of Code (GSoC), and two undergraduate women of color on their computer science careers through the Computing Beyond the Double Bind (CBDB) Mentoring Network. I worked with the junior graduate student on an extension of my research, and coached him on presentation and writing techniques. The GSoC students worked remotely on enhancements to my open source research projects; I communicated regularly with each through video chat, email, and GitHub. The CBDB Mentoring Network connects computing professionals with undergraduate women of color in computer science; I discussed graduate school, job interviews, and professional issues with my mentees over email and video chat.

While mentoring a student on a research project, I work with the student to form a project roadmap with realistic and measurable goals. The roadmap is not intended to fix a timeline in stone; instead, the process forces students to think through an idea, identify untested hypotheses, and determine metrics of success. When a student encounters a roadblock, we work together to adjust the roadmap and determine a new course of action. One undergraduate had to completely change his project after a few weeks when a goal turned out to be infeasible; we adjusted the roadmap and he later completed a new project successfully.

Teaching Interests

I would love to teach undergraduate courses on web programming, software security, software engineering, programming languages, operating systems, and networking. These courses cover topics that are essential to modern web application development, which is the focus of my research. At the graduate level, I would like to teach a project-based course in which students present and apply recent ideas in systems research to new domains. From my experience in a similar course, the process of understanding, adapting, and implementing someone else's compelling idea provides a deep appreciation for the subtleties in others' work.

[2] Students within the Five College Consortium can take classes at UMass. Specifically, students from Mount Holyoke College and Smith College enrolled in the course.