Bor-Yuh Evan Chang

Teaching Statement

Teachers open the door. Students enter by themselves. — Proverb

Overview

The quote above sets the tone for my teaching: a teacher's primary role is to create a simultaneously enriching, challenging, and compassionate environment that supports the learning and growth of the individual student. This fundamental principle underlies both my classroom teaching—as exhibited by adopting a project-based approach for my courses—and my individual mentoring activities—as exemplified by my use of daily stand-up meetings.

Since joining the faculty at the University of Colorado (CU), I have taught courses across the graduate and undergraduate curricula. At the undergraduate level, I have invested in a significant refresh predicated on project-based and active learning principles for CSCI 3155 (Principles of Programming Languages), a foundational course required by all Computer Science majors. Without any explicit advertisement, a colleague at Tufts expressed interest in adopting my materials from this course, as well as a professor from National Chengchi University in Taiwan. I have also contributed to the improvement of CSCI 4555/5525 (Compiler Construction), an advanced undergraduate elective also commonly taken by master's students. At the graduate level, I have worked on revamping CSCI 5535 (Fundamentals of Programming Languages), an introductory graduate course to the formal semantics of programming languages. I have also developed a course on Program Analysis (CSCI 7135) that both strengthens the department's graduate offerings on formal methods and complements my research activities. The success of the first iteration of CSCI 7135 in Fall 2010 led to a request to offer a follow-on, advanced practicum course on the subject in Spring 2011. I discuss my approach and work on these classes in greater detail below.

With respect to outreach, in what was a particularly fun event, I led a "What is Computer Science?" workshop as part of the ASPIRE Summer Bridge program for newly admitted first-year engineering students organized by CU's BOLD (Broadening Opportunity through Leadership and Discovery) Center. The goal of the workshop was to dispel any misconceptions about the study of computer science by putting forth my assertion, "Computer science is about solving puzzles with social impact."

I have graduated three Ph.D. students, placing them in strong teams at the Center for Computing Sciences at the Institute for Defense Analyses (Arlen Cox), Apple (Devin Coughlin), and Facebook (Sam Blackshear). They have made highly visible contributions after graduation. For example, Coughlin was behind a highly advertised feature of the Swift 2 programming language—the feature was discussed at Apple's Worldwide Developers Conference 2015. I have also graduated four B.S. thesis students, am currently mentoring two postdoctoral research associates, three Ph.D. students, one B.S. thesis student, one M.S. student, and two B.S. students on independent research. In total, I am (or have been) the primary advisor for six Ph.D. students and five B.S. thesis students, as well as working with five M.S. students and six B.S. students. Of these eleven B.S. students, six of them started in my group as Discovery Learning Apprentices, a program through the College of Engineering and Applied Sciences.

As a faculty advisor, I see my role as amplifying the efforts and facilitating the success of those around me, beginning with my Ph.D. students. Of my twenty-six publications since joining CU, fifteen have been with my direct advisees and in total twenty-two have been with student co-authors from CU and elsewhere. Two of my Ph.D. students have won prestigious graduate fellowships: Chateaubriand Fellowship in Science, Technology, Engineering, and Mathematics from the Embassy of France (Arlen Cox) and a

Facebook Graduate Fellowship (Sam Blackshear). I have also placed Ph.D students at top research labs for summer internships, including Microsoft Research Redmond, Cambridge (UK), and India, NEC Labs, IBM Research Tokyo, Apple, and Facebook.

Integrating my teaching activities with my research activities has been an important way to recruit students into my group. All five B.S. thesis students along with three other B.S. students became interested in programming languages research through their experiences in my offerings of CSCI 3155.

I spend a significant portion of my energy in organizing the CU Programming Languages and Verification (CUPLV) Group[1] to create a collaborative and supportive environment that fosters excellence including but also beyond my own advisees. Currently with Profs. Sriram Sankaranarayanan, Pavol Černý, and Matthew A. Hammer, we seek to create a collaborative and supportive community with a critical mass that would be difficult to achieve within any individual faculty member's advisees. I credit the strong community among our graduate students for the successes of our research group at least as much as the individual faculty mentoring. I also credit the group in our ability to attract strong Ph.D. students and postdoctoral research associates. The recruiting environment in computer science has become particularly competitive for potential Ph.D. students and postdocs because the competition is not only with other universities but also with technology companies offering six-figure entry-level salaries.

In the following, I describe in further detail the application of my teaching philosophy, the challenges that I have faced, and the strategies that I have employed to improve.

Developing as a Teacher

My development as an instructor has benefited from leveraging the resources and workshops offered by the Faculty Teaching Excellence Program (FTEP). In Spring 2010, I took part in the "Teaching in a Nutshell" workshop facilitated by Prof. Lee Potts. This experience was invaluable in shaping and ultimately improving my classroom style. I have also attended FTEP seminars on the first day of class and the changes coming with on-line lectures and courses. When I taught my first large, 100+ student class, I received guidance and encouragement through the video consultation service, and I have followed up by engaging in the Classroom Learning Interview Process (CLIP) in Fall 2015. These experiences are part of my effort to continuously reflect on and improve my teaching. I have taken the opportunity to discuss and sit in on the classes of my colleagues. These experiences have informed how I have structured my courses, including abandoning the use of PowerPoint slides in favor of a conversational classroom and the introduction of short, frequent quizzes to provide students with prompt feedback on their level of understanding. The conversational style has led to deeper student engagement, and frequent quizzes have helped students better assess their progress. I attended a very helpful workshop organized by the College of Engineering and Applied Science and facilitated by Prof. Michael Prince (Bucknell) on active learning in the same semester in which I was revamping CSCI 3155 (Principles of Programming Languages at the undergraduate level), as well as a recent workshop on graduate advising organized by the Graduate School.

Classroom Teaching Philosophy

Reflecting on the courses from which I learned most and the instructors who I admire most, I have observed that quality teaching begins with a recognition of different ways of learning, deliberate and careful organization of learning goals, and a genuine enthusiasm for both the subject matter and the act of sharing it. While definitely challenging, I strive to create such an environment for *all* students.

[1] CU Programming Languages and Verification: `http://plv.colorado.edu`.

Project-Based Course Design and the Conversational Classroom. Following the principle that my job is not to deliver tidbits of knowledge but rather to create an enriching environment for learning, I have embraced project-based courses and active learning ideas. With the design of every course, I begin by first making clear the learning goals of the course. These goals then drive the development of *hands-on homework projects* that students tackle through the course of the semester. The classroom meetings are then driven by dialogue about the thinking and background needed to tackle the project. As a concrete example, my undergraduate courses are typically structured into a sequence of two-week homework projects, each with clear and definitive learning goals. My lecture preparation consists of understanding the typical stumbling blocks in the project and creating discussion material to guide students past these stumbling blocks. The content of a lecture is never set but instead is driven by the questions that the students have on the project that day. While the discussion points for a given two-week project are the same from semester to semester, the order in which those points are covered is typically different. I have found this approach extremely effective in driving an interactive, engaging classroom.

The most common positive feedback that I hear from students on my courses are how much they learned and how much they were challenged by the homework projects. My impression of the reputation of my courses is that students will need to work very hard but if they do, they will get a lot of benefit from the experience.

> *I have never been so challenged in a course, and I am thankful for it.*
> *— Student in Fall 2015 CSCI 3155*

Continuous and Personalized Feedback. I have found a particularly challenging aspect for students in a project-based course is figuring out if they achieved the learning goals—whether they have been able to synthesize and internalize the higher-level concept from the detail work on the projects. To address this concern, I have employed three different strategies: (1) in-class exercises, (2) on-line drills, and (3) interview grading. In the subsequent class meeting after a project deadline, I administer an exercise or "quiz" consisting of questions covering the learning goals of the project and representative of exam questions. I create on-line drills to offer practice in basic skills and feedback on conceptual understanding. Finally, homework projects are graded through an interview process with a teaching assistant. I prepare a rubric and a set of questions for the teaching assistants, and the purpose of the interview is to uncover any misunderstandings the student might have about the project in an individualized setting. I have received feedback that the interview process, while intimidating at first, is one of the most helpful aspects of the course.

Continuous Monitoring. One of the difficulties with a heavy emphasis on homework projects is making sure that the course load is appropriate. One common issue is that students are sometimes unclear about the deliverable and spend more time delivering something beyond what I had expected of them. While I encourage interested students to explore, I also understand that students are balancing many obligations and should have a work load commensurate with the credit hours for the course. To address this issue, I include a survey on each assignment where I ask students to estimate the number hours spent, a qualitative assessment of the difficulty, and comments on the parts that gave them the trouble. I then review the results in a subsequent class to address any disconnect in expectations and to emphasize my commitment to a reasonable work load.

Research Mentoring Philosophy

Beginning from the foundation that my role is to facilitate and engender the success of my advisees, I have developed several strategies to promote a transition to independent thinking while providing a

structure and foundation for that development.

Providing a Structure and Foundation for Research Independence. With any student (or postdoc), I discuss what I believe it means to be a successful Ph.D. student (or undergraduate researcher, etc.) and ask them about their goals after graduation. I explain that my advice may differ depending on their goals and that I wish to stay connected with respect to their goals as they might change. With a new Ph.D. student, I emphasize getting active in research from day one—not after taking classes. I typically have them get started on an ongoing project with another student where they are expected to begin contributing on all aspects (brainstorming, implementing, writing, revising) right away to the best of their ability. Simultaneously, I encourage them to start thinking about how they can transition to more of a leadership role where they are the clear owner of a project.

Driven by the goal of supporting students while promoting ownership of their own research, I have continuously tweaked the way I conduct research meetings with students. I no longer have regularly scheduled weekly meetings with each student as is common—instead, I meet with them much more frequently in a much more fluid manner. Each day, the most important meeting of the day for me is a fifteen minute stand-up meeting with all of my students together to review their plan for the week. It is at this meeting that I can assess how a student is progressing and then schedule lengthier research meetings. This process helps me maximize the times I reserve for student research meetings: longer or shorter discussions as beneficial to the student.

Fostering a Collaborative and Supportive Community. I have also placed a strong emphasis on community, understanding the significant impact a dynamic group of peers can have on the success of the student. To support this effort for our collective students in the CUPLV group (across multiple faculty advisors), I organize a weekly student seminar series, a weekly reading group, and a twice-weekly, "stand-up status meeting." Each Ph.D. student is asked to give one slide-based presentation and one whiteboard-based presentation each semester in the seminar series. The stand-up status meetings have been fantastic for getting students into the lab, connected, and helping each other. I also run "critique meetings" where we get together—as a group—to critique and offer constructive suggestions on paper drafts and presentations.

While technical skills to solve research problems are important, I place even greater emphasis on *justifying* that they are addressing the right problems. I challenge students to continuously refine their problem statement and their justification of why that problem is *important*, *hard*, and *interesting*. I take every opportunity to support this thinking collectively as a group, including for example a group meeting devoted to delivering "elevator pitches" to be prepared when meeting important people at conferences.

Graduate Courses

Previously, the graduate offerings in the programming languages were not coordinated, resulting for example in three graduate courses being offered in one semester and none in the next. I organized an effort to coordinate schedules such that (1) we balance the graduate course offerings in this area across semesters and (2) schedule the foundational courses in the Fall and the more advanced courses in the Spring.

At the graduate level, I have alternated between teaching the foundational course on Fundamental Concepts of Programming Languages (CSCI 5535; in Spring 2009, Spring 2010, and Fall 2013) and a specialized course on Program Analysis (CSCI 7135; in Fall 2010, Spring 2013, Fall 2014, and Spring 2016). CSCI 5535 serves to provide a solid foundation to students going into programming languages research but is also a breadth course for students in other research areas. CSCI 7135 is a specialized graduate course. While many students are those engaged in the CU Programming Languages and Verification

group, there are often a few other graduate students with a secondary interest in developer tools. Advanced undergraduates are also welcome: a few undergraduates take the course each semester. In this course, I have structured the course project in a unique manner that is both collaborative and independent. Each student (or pair) undertakes a research project of their choosing, but all students contribute to a common infrastructure for building their analyzers. Particularly in the Fall 2010 version, this collaboration led to a lot excitement around the project. So much so that some students lobbied me to offer a second semester of the course to continue the project, which I obliged by offering a practicum course in Spring 2011 on top of my normal teaching duties.

Video Lectures and Distance Learning. I have offered most of my graduate courses through the distance learning program at CU. This program provides distance learning to working professionals by offering recordings of on-campus classes on-line. With distance students, the most significant challenge I have faced is managing the lower engagement level, particularly given my conversational classroom. To address this, I have offered "on-line office hours" via instance messaging, phone, or video conferencing. Then, in all of my classes (distance or not), I record the classroom discussion on a tablet computer (instead of on the blackboard) so that I can post pdf "whiteboards" after class. I have also promoted heavy use of class-wide on-line discussion forums, which has been beneficial for both distance and in-class students. I was one of the earliest adopters and one of the heaviest users of the on-line engagement tool Piazza at CU. I first used this technology in Fall 2011 and students liked it so much that we generated 1,253 posts.[2] I believe this challenge in engaging with distance students contributes to lower instructor ratings in these distance sections compared to the on-campus sections (3.7 versus 5.4 in Spring 2012, 4.3 versus 4.7 in Fall 2011, 4.3 versus 5.4 in Spring 2015), though these averages may also have been affected by low enrollment with sometimes one or two disengaged students (e.g., of the three distance students from Spring 2010, two rated the course with a 1 or a 2 but one student gave a 6 rating). While the trend is upwards, I continue to work on improving engagement with distance students.

Undergraduate Courses

My primary effort in undergraduate education has been developing and teaching CSCI 3155 (Principles of Programming Languages). I have taught this course five times in Fall 2009, Spring 2012, Fall 2012, Spring 2014, and Fall 2015. This course is both theoretically foundational and practically oriented. The programming languages in use are constantly changing, so a highly-qualified software engineer must be able to adapt to new languages and technologies quickly. This course elucidates the underlying principles of programming languages and computational models so that one can easily see how a new language is an evolution from existing ones. I am very explicit about this goal to the students: I say that the primary goal is to help them become "non-outsourceable software engineers."

Experience Revamping CSCI 3155. In Fall 2009, I followed closely existing materials for the course while having many discussions with students on how to improve the course. Based on that experience, I undertook a revamping of the course in Spring 2012 following project-based principles. While the pedagogical research and my own anecdotal experience have shown that students are much more engaged and end up more likely to understand and retain the subject matter using project-based approaches, I was surprised by the difficulties and resistance that I encountered that semester. This was in the end reflected in the lowest FCQ instructor rating in my pre-tenure period (3.7), whereas my average instructor rating in all other on-campus courses is 5.1. Fortunately, I was able to seek the support and advice of my senior colleagues, including one of our department's Presidential Teaching Scholars. Based on those discussions, I saw there were a confluence of factors and mistakes that led to this outcome. First, I severely underestimated the difficulty in addressing the needs of a large, 100+ student class in contrast

[2] Courses at CU using Piazza: `https://piazza.com/school/colorado/`.

to the ~40 student classes I had taught previously, and this was magnified by my discussion-oriented style. Second, a project-based course requires setting expectations for a certain level of independence. Third, this course is the last required course for computer science majors, and thus unfortunately there are some students who just do the bare minimum to graduate. Finally, I allowed the extreme negative feelings of one student to spread through the class even though they were not universally held. Given this experience, I asked to teach the course again in the next semester (Fall 2012) and took this experience into account. In particular, I was conscious about making clear the learning goals, the reasons for the project-based course, and the expectations required of them. I also added quizzes to help students better assess their own progress, and I paid close attention to students who I thought were falling behind by inviting them to meet with me. This resulted in a significant FCQ instructor rating improvement to 5.3. Note that both semesters had students who were extremely positive about the class: 3 students later joined my research group, and 3 others asked me for letter of recommendations. Multiple students have come by later to tell me how much they appreciated what they learned in the class, including one student from Spring 2012 who was rather unhappy during the semester. Overall, I have been happy with this turnaround without sacrificing the goals I had set out with incorporating project-based principles. According to Prof. Michael Prince, an expert in active learning techniques from the workshop I attended, this initial negative reaction from students is quite common for courses transitioning to project-based techniques.

What has crystallized in the subsequent semesters is that I must set expectations about the project-based structure clearly at the beginning of the semester. I must be clear that the course is technically deep and challenging, but at the same time, the course is structured so there are many opportunities to get past stumbling blocks before taking the exams. My project-based course design philosophy is largely informed by my experiences in developing this course.

Future Work. Grounded by my teaching philosophy, I continue working to improve as an instructor through self-reflection and foremost listening to students. To do so, I will continue to follow my existing approaches for surveying students, as well as finding new ways to get feedback (e.g., through FTEP's CLIP). I am actively searching for ways to offer students more continuous and personalized feedback through, for example, videos, interactive and guided exercises, and other hands-on mediums, as I believe doing so has the most potential for improving student learning.