Teaching and Mentoring Statement

Caleb Stanford

Fall 2021

My interest in teaching has been a primary factor in my decision to pursue an academic career. Throughout my time at Penn, I have taught the undergraduate Rust Programming course in spring 2021, mentored junior PhD and undergraduate students, mentored local high school students in advanced placement (AP) computer science, and served as a graduate TA. For my efforts in teaching, I earned a teaching certificate from the Penn Center for Teaching and Learning (CTL). I believe in an evidence-based approach to teaching, particularly with regard to effectively teaching underconfident and academically at-risk students. My personal experience is that engaging teaching and exposition not only help students, but also lead the teacher to a more robust understanding of their field.

Teaching Experience. In Spring 2021, I taught Rust Programming (CIS198, ∼16 students), an undergraduate course focusing on systems programming, for which I developed a great deal of new course material, which is publicly available on GitHub. I took this opportunity on top of the usual teaching load for PhD students (2 semesters of TAing) because I wanted to sharpen my exposition of fundamental programming languages and systems concepts. One thing I brought to the course was an increased focus on developing data structures in Rust, which can be notoriously difficult due to the requirement to deal with smart pointer types and sound memory management; I expanded this into several lectures, and multiple students told me that my lecture introducing the reference-counted pointer (`Rc`) and dynamic ownership (`RefCell`) was their favorite lecture of the course. In addition to the students registered, I was fortunate to have enthusiastic participation from a few PhD students auditing the class, as well as recurring emails throughout the semester from external people looking to follow along with the online material or videos, or even to re-use it for their own teaching. In my course evaluations, I received Very Good ratings (3 out of a scale from 0 to 4) for overall quality of the instructor (3.2), overall quality of the course (2.9), and a number of other subcategories.

Teaching the course virtually during the pandemic increased the difficulty of keeping students engaged. I believe that while on the one hand some students prefer to go through material asynchronously, many others heavily benefit from some form of interactive learning. To balance these concerns, I iterated over different strategies to mandate engagement in lightweight and open-ended ways. For example, one effective strategy was mandating that every student ask a question on Piazza each week for me to answer or discuss in class. This both helped students stay up-to-date with the material and helped me realize when students were struggling with some concepts. Another technique I found to be worthwhile, and not always prohibitively time-consuming, was to collect student work using GitHub and provide iterative feedback using pull requests with comments. This emulates the way software engineering is done in practice, whether in a small team or at a large industrial company, and is far more useful for students than only providing feedback at the time of grading. Multiple students called out this method of feedback and thanked me, both by email and in the course survey.

Besides teaching Rust, I spent a fruitful year of TAing under Prof. Benjamin Pierce for the programming languages and verification course Software Foundations (CIS500, ∼50 students), and under Prof. Sampath Kannan for Theory of Computation (CIS511, ∼50 students), totaling 10-20 hours per week. I was praised for my dedication as a TA. For both courses, I was the most active instructor in providing detailed answers to questions on Piazza and was closely involved in preparing homework and exam questions. I also made lasting contributions to the autograding infrastructure distributed with the *Software Foundations* textbook.

Teaching Interests. At the undergraduate level, I am excited to teach classes in (1) programming and compilers, (2) systems programming, (3) concurrent and distributed systems, (4) theory of computation, and (5) logic in computer science. I would also be interested in teaching undergraduate- and graduate-level courses on automated formal verification and big data analytics systems. Finally, I could teach graduate seminar courses in areas related to my research, e.g. SMT solving, dataflow programming languages, and advanced formal logic.

I am interested in introducing formal tools and formal logic into the undergraduate curriculum. For example, in mid-level to advanced programming courses, software verification tools including Dafny could

be introduced as a way to validate software design and correctness. For systems programming, Rust offers novel ways to teach about type systems and memory management. Other logic tools, such as the Spin model checker, can be valuable aids in understanding concurrent systems. A full-fledged course proposal in this direction would be Logic for Systems, based on Prof. Tim Nelson's course at Brown, my undergraduate institution. On a personal level, Logic for Systems inspired me to pursue research on applications of logic in programming languages and systems.

Mentoring. In mentoring, I am guided by the principle that I should always be *available*: available for advice, to listen to and share experiences, and to help resolve technical and emotional roadblocks. At Penn, as a senior PhD student I have been a mentor for four junior students on my projects over the last four years; I have cherished the many conversations and sessions with these students, not just on research projects but discussing their research and career goals, roadblocks they have faced, and how to navigate the environment at Penn. I am also currently supervising a wonderful undergraduate thesis project. In addition to my opportunities at Penn, I have been a long-term mentor in the SIGPLAN-M program for an early PhD student, and a mentor for URM PhD applicants in the 2020-2021 application cycle. This semester, I was able to connect my SIGPLAN-M mentee with a potential industry internship opportunity. I try to extend the same availability to those outside of my immediate circles: for example, I always eagerly volunteer for student panels and visit day activities. I love hearing students' questions and trying to give the most honest and helpful answers.

Outreach. I am a co-founder and problem-writer for the Utah Math Olympiad, a proof-based high school mathematics contest for motivated high school students in Utah. As a current outreach goal, I am interested in high school programs which make computer science more accessible at the undergraduate level. Current high school CS programs are too often focused on learning Python or JavaScript; I believe we need more ambitious programs which focus more on fundamental CS topics like algorithms and data structures, and which develop mathematical maturity. Currently, I am incorporating some of these outreach teaching ideas in my involvement with the Steppingstone Scholars high school AP CS outreach program in Philadelphia. The goal of Steppingstone Scholars is to teach CS to underserved schools, and to prepare them for the AP CS exam in the spring. My personal stake in the matter is that I am trying to supplement the (asynchronous and already present) curriculum with more concept-based programming challenges during class time.

Teaching Philosophy. *I believe in an evidence-based approach to engaging and effective teaching.* In the course of teaching at Penn, I have taken the opportunity to reflect on what I can do to improve. To evaluate myself objectively, I met with a CTL fellow one-on-one, who observed my teaching and identified specific suggested changes. For example, we identified together that starting each lecture with an outline or roadmap can help orient students and make the lesson easier to follow; since then, I have incorporated a clear, accurate lesson plan in every lecture. I also attended and actively engaged in several workshops from CTL. What particularly stands out to me is the Fall 2020 online mini-course, taught by Prof. Cathy Turner, primarily due to her engaging discussion- and interaction-oriented approach. She emphasized having instructors engage with each other and share experiences, ideas, and course lesson plans. Through this workshop, I was able to make important choices in advance about my planned teaching material for Spring 2021, including having a motivating but lenient late policy for homework, and mixing synchronous and asynchronous material.

As a final experience and memorable example of evidence-based teaching, when I was TAing for CIS500, it had been decided that we would grade the exam differently than in previous years, by penalizing guessing. This concerned me, because it has been shown that penalizing guessing sometimes more adversely affects grades of women and minority students. The research understanding for why this likely occurs is that taking points off for guessing penalizes underconfidence: despite being equally competent with their peers, underconfident students will be less likely to guess even when their answer would be correct. I presented these research findings to the professor and convinced him to reverse the planned change in policy. I still feel a deep respect for the professor's decision to respond to actionable evidence.

As a teacher, I hope to live up to this ideal of integrity and humility. I will continually adapt my teaching and policies to be more effective, to incorporate morality and ethics where appropriate, and to respond to societal inequalities. I look forward to implementing these principles in my future role as an academic.