Teaching Statement

Joshua Gancher

I am very excited to continue my career as a teacher and mentor, in addition to doing research. My experiences with teachers throughout my computer science education, as well as my own experiences teaching and mentoring, have solidified my philosophy: **put the person first**. Every student has a different motivation for being here, and it is my job — in teaching coursework, and serving as a mentor to graduate and undergraduate students — to meet them at their level.

Teaching Strategies My person-first attitude towards teaching has a number of direct implications for how I believe classrooms should be run. Foremost, every student is different; indeed, in a single classroom, one is guaranteed to find a large diversity of academic backgrounds, motivations, learning styles, and communication strategies. Computer science coursescannot be effective without paying explicit attention to this diversity.

Indeed, the best teaching strategies are *multimodal*. For example, by using visual representations in slides in conjunction with textual readings, students are much more likely to "connect the dots" than via only one of these teaching strategies. Similarly, the pandemic has shown us that some students learn better in synchronous settings, which others prefer asynchronous ones.

Additionally, the best classroom experiences are those with *multiple ways of engagement*. The best classes are those which blend theory with practice. By having assignments that practice *both* theoretical and practical skills, I will engage students with a variety of interests, and help them synthesize a broad view of the material.

Mentoring Strategies In mentoring both undergraduate and graduate students, the most important thing is to communicate that *this time is for them*. As such, I plan to not tell students exactly what to work on, but rather help them find interesting problems that I am able to coach them through. By giving students the agency to form their own research agendas, I aim to help them take ownership over their research.

Additionally, doing research — particularly research in computer science — is fundamentally a *creative process*. While one should encourage students to achieve their full academic potential, I believe that the best creative work is done with a healthy work-life balance. As a graduate student, I made the all-too-common mistake of working too much, only to get not enough done. Inventing new, fundamental solutions to hard problems takes ingenuity; without a healthy dose of cognitive rest, such ingenuity is much more difficult to attain.

Teaching and Mentoring Experience As a graduate TA, I have assisted in teaching two undergraduate classes at Cornell: systems and architecture (CS 3410), and compilers (CS 4120). Via office hours, I have gained significant experience in direct one-on-one teaching; additionally, via evaluating student work in large classrooms (over 200 students for 3410), I have gained practical experience in how to efficiently guide students and other TAs in these settings.

As a postdoc at Carnegie Mellon, I have had the pleasure of mentoring a variety of PhD and M.S. students, resulting in a joint publication in IEEE S&P 2023 and upcoming submissions to ASPLOS 2024 and IEEE S&P 2024. Transitioning from PhD student to postdoc has shown me the great satisfaction to be derived from helping students grow and succeed as researchers. Indeed, throughout the postdoc, my favorite parts of the day are those spent assisting others through one-on-one meetings, pair programming, re-scoping projects to fit the student's skills, and occasional long chats about what they want to get out of grad school.

Additionally, in the 2022 academic year, I served as an undergraduate thesis advisor for a student from my own undergraduate institution, Reed College. The student had a burgeoning interest in programming languages, and after discussion of many topics, we landed on the topic of logical relations. While this topic has a reputation for being quite difficult, my impression was that this perceived difficulty can be mitigated by a well-designed learning plan. Correspondingly, I worked with the student through a semester-long curriculum which successfully took them from a basic knowledge of lambda calculus to a full grasp of logical relations for the security-typed dependency core calculus. They achieved the highest marks for their thesis, and deservedly so. In addition to their remarkable dedication and persistence, my success as their mentor stemmed, I believe, from my person-first attitude. By explicitly *not* talking about work at the beginning of each meeting — and asking how their day is going — we established a mutual trust upon which a collaborative teaching experience can be built.

Relevant Topics As a researcher of both **cryptography** and **programming languages**, I am well qualified to teach both of these subjects at the undergraduate and graduate levels. On the programming languages side, I can teach courses on a wide variety of topics, including functional programming, program logics and verification, type systems, language-based security, and formal methods. On the cryptography and security side, I am well equipped to teach the topics of communications security, software security, side channels, complexity theory, and modern cryptography including public/secret key cryptography, multi-party computation, homomorphic encryption, and zero-knowledge proofs. Additionally, I am interested in teaching foundational topics such as mathematical foundations, proof-writing, and discrete structures.

I am particularly interested in developing two new, hands-on classes: **formal verification**, where students learn about using tools such as Coq and Dafny to verify software systems; and **computer-aided cryptography**, where students learn about more specialized tools to verify cryptographic software, apply side channel analyses, and mechanize proofs of cryptographic security. Both classes would be great for advanced undergraduate and graduate students, and would involve substantial theoretical components along with lots of hands-on experience.