

My teaching philosophy is to help students become active and independent life-long learners. Toward this goal, I have endeavored to create an interactive and inclusive environment during my teaching sessions and through outreach activities.

Teaching Experience. During my graduate studies at the University of Illinois Urbana-Champaign, I worked as a teaching assistant (TA) for CS427 Software Engineering (I) in the Fall 2019 and Fall 2020 semesters. The course had around 100 students in 2019 and 190 students in 2020. It covered basic concepts of software engineering practices, environments, and processes. My duties included reviewing and grading students' assignments, preparing and maintaining materials on the course website, hosting weekly office hours, providing hands-on assistance, and supervising students on their final team projects. I served as the lead TA in 2020 and was responsible for hosting weekly TA meetings, coordinating with the instructor to assist in daily teaching plans, and monitoring administration-related questions. Besides, I gave guest lectures in CS422 Programming Language Design, CS522 Programming Languages Semantics, and CS598 Runtime Verification on various topics in category theory, functional programming, runtime verification, and my Ph.D. research.

I am eager to use **student-centered teaching methods** and help my students become independent learners. For example, in CS427 Software Engineering (I), I used the inquiry-based learning method to supervise students' team projects, where students themselves proposed use cases to be implemented for the projects, developed timelines and set weekly goals, and allocated teamwork. Instead of a lecturer, I played a facilitator role who participated in their discussions and provided only guidance, not direct answers. Sometimes, a team could be too ambitious and propose too many use cases. Instead of directly telling the group that the proposal was too big, I encouraged them to divide the workload and set weekly goals. After that, the students realized that the workload was too big and adjusted the proposal accordingly. I found this technique especially effective in supervising team projects because students viewed me not as an outsider but as their partner, making my advice and guidance more effective.

I have experience in **online teaching**. In the Fall 2020 semester, CS427 Software Engineering (I) was delivered entirely online for the first time, and the course structure was significantly changed. As the lead TA, I actively coordinated with the students, the instructor, and the other TAs to ensure that the online course format was effectively implemented. For example, the online course used an existing Coursera package with pre-recorded lecture videos, quizzes, and assignments instead of live lectures. On the other hand, the online course inherited the final team projects and other reading materials from the past face-to-face version. I helped merge the Coursera package and the face-to-face course into a cohesive and consistent one. Before the semester began, I helped the instructor adjust the Coursera course agenda to fit the new schedule of the semester and added new Coursera pages that linked to the face-to-face content. I worked with the other TAs to modify the face-to-face contents to reflect the new online course format. During the semester, I helped upload Zoom recordings of the instructor's office hours, supervised students' final projects on Slack, and monitored students' administrative questions on Salesforce.

I pay special attention to promoting **education inclusion in hybrid flexible (HyFlex) teaching** and ensuring that remote students have an equally engaging experience as in-person students. In CS522 Programming Languages Semantics, I gave a guest lecture on my Ph.D. research. The course was in a HyFlex format, with some students attending in person and others joining remotely. To create an engaging environment for all students, I arrived at the classroom 10 minutes before the class began to make small talk with my students, both in-person and remotely. It not only created a welcoming environment, helping them to get to know me better, but also allowed me to test the camera, the

microphone, and slide sharing. In class, I made eye contact with students in the classroom and looked into the camera to make “virtual eye contact” with remote students. When I asked students whether they had a question, I said, “do we have any questions?” and then, “do we have questions from chat?” to ensure that the remote students were also involved in the discussion. I repeated questions from the classroom so the remote students could hear them well.

Mentoring Experience. During my graduate studies, I had the honor to serve as a research advisor two undergraduate students, three master’s students, and three Ph.D. students from the U.S. and abroad. One undergraduate student published a top-tier conference paper with me, with another paper in the submission pipeline, and is now pursuing a Ph.D. at Carnegie Mellon University. One Master’s student recently graduated, and I served as an external consultant for his thesis.

I dedicate my time and effort to building mentor-mentee relationships. I hold one-on-one weekly meetings for each student to discuss their studies and research and use Slack for offline discussions. My general mentoring objective mirrors my teaching philosophy: I seek to establish a learning relationship that helps students to take charge of their development and achieve their goals. As for concrete mentoring methods, I have been learning by reflecting on my personal experience as a mentee with my Ph.D. advisor and reading articles that offer guidance for mentors. For example, I borrow the idea of **demonstrations** from active learning strategies in my mentoring. Through my actions, I prefer to show students the best practices for conducting quality research. When I comment on a student’s draft paper, I give critical feedback and make concrete, implementable suggestions as comments in the PDF files. Although it takes me more time, it is a good learning experience for the students to merge and implement the feedback. It also gives the student a sense of ownership of their paper and makes them feel valued for their dedication.

Outreach Activities. For me, teaching is not limited to lecturing in a classroom. I have consistently sought opportunities to break knowledge barriers and engage a larger audience. For example, I gave a public tutorial at ICFP 2020 (<https://shorturl.at/ovCPI>), a top-tier conference in functional programming. In the 3.5-hour tutorial, I presented how to use the K framework, a tool from our lab, to formalize functional programming languages. To engage the audience, I included demos and hands-on sessions for the participants to try our tool. I ran the tutorial with a mix of lecturing and discussion and encouraged the audience to ask questions. A recording of the entire tutorial is available online (https://www.youtube.com/watch?v=VIQMj_N42B8) and has gathered around 1000 views.

In the summer of 2022, I helped the startup Runtime Verification Inc. form its new research arm, RV Research. I helped create the RV research website (<https://research.runtimeverification.com/>). I recorded introductory videos on several open problems with practical impact to bridge the gap between academic research and industrial interest.

Teaching Interest. Based on my teaching and research experience, I am qualified and eager to teach graduate classes related to formal methods, programming languages, and software engineering. Such courses will allow me to share my academic and industry experience in using formal and informal methods to improve the safety and reliability of computing systems. Formal methods courses will allow me to explore semantics-first techniques and tools for building reliable software. Programming languages courses will let me discuss major language design paradigms using imperative, functional, and logic programming as unifying themes based on my expertise in language semantics framework. Logic courses will allow me to give students the big picture of the

usage of logic in computer science, help them understand connections between logic and algorithms, and prepare them for using logic tools to solve problems.

Beyond teaching existing classes, I also like to create new courses that bring together traditional topics in programming languages and formal methods and new computing scenarios. For example, my industry experience and research make me an ideal candidate to teach a course on programming languages and blockchain, which will explore the design, implementation, and formal analysis support of programming languages through the lens of blockchain applications and smart contracts. I also look to organize seminars on programming languages and formal methods. Overall, I will help students at my institution to succeed in academia and industry.