Teaching Statement

GUANNAN WEI

I value teaching as an important way to disseminate knowledge. My main passion in teaching lies in subjects related to programming and programming languages. Throughout the teaching process, my goal is to help students appreciate intellectual ideas and develop practical problem-solving skills to make effective use of computing. Furthermore, I am enthusiastic about mentoring students at all levels, guiding them on their journey toward becoming future researchers or engineers in academia and industry.

TEACHING & MENTORING PHILOSOPHY

- ***Learn by Doing.*** I firmly believe that students acquire knowledge most effectively when they immerse themselves in the material through hands-on activities, projects, and real-world applications. To understand how things work is to construct them – this is especially true for both programs and proofs. By enabling students to translate theoretical concepts into practical scenarios using in-classroom examples and projects, my goal is to amplify their comprehension and long-term retention of the subject matter by deploying the active learning theory.
- ***Teach the First Principles.*** Understanding the intellectual ideas behind complex subjects is crucial to develop skills for students to solve complex problems. Rather than approaching topics as "black boxes", I believe it is essential to teach the "first principles" and demonstrate a bottom-up reasoning path or construction that begins with minimal assumptions. From my past teaching experience, this is often an engaging process that demystifies intricate concepts and enables students to apply what they have learned to solve new problems.
- ***Encourage Curiosity and Creativity.*** Fostering intellectual curiosity and nurturing creativity in students is at the core of my mentoring approach. I aim to create an environment that not only encourages students to seek solutions but also provides the freedom to ask "what-if" questions and explore consequences. Sometimes, to be creative is to play. By promoting an atmosphere of curiosity and inclusion, I hope to inspire innovative thinking and a genuine passion for learning.

TEACHING EXPERIENCE

I served as a teaching assistant (TA) on four occasions for two distinct courses. In Fall 2019 and Spring 2020, I was a TA for the graduate and undergraduate courses on *compiler construction*, respectively. In Spring 2018 and Fall 2017, I served as a TA for the undergraduate *system programming* course. Additionally, I have often given guest lectures on topics related to compilers and static analysis for both undergraduate and graduate students.

Lead Teaching Assistant, Principles and Practice of Compilers. Compiler construction techniques are one of my favorite topics to teach. In this course at Purdue, we teach students in compiler construction starting from *language semantics* and progressively guide them to build an optimizing compiler for contemporary languages through a series of projects. It is a demystifying moment for students when they realize that compilers are, after all, just programs that translate other programs, informed by the language semantics.

The course projects are deliberately designed to be self-contained, not relying on pre-existing tools/frameworks (e.g. Flex/Bison and LLVM). Yet they still cover the most important and engaging topics, such as X86-64 code generation, type checking, optimizations and transformation over intermediate representations, etc. While employing an off-the-shelf compiler framework may offer convenience, our emphasis is on providing a more pedagogical experience that focuses on the underlying principles. In this way, students gain a deeper and more enduring understanding that will prove valuable in their careers beyond graduation.

I enjoyed my TA experience, in particular, explaining intricate concepts by relating them to students' prior programming knowledge. For instance, to explain the concept of continuations, I often crafted simple Python programs for students to demonstrate how to use functions to encode control flow. Additionally, I provided interactive code debugging sessions that closely helped students, particularly at the undergraduate level. This interactive approach proved effective in pinpointing gaps in their mental models when dealing with problematic code.

The Spring 2020 semester posed particular challenges with the abrupt shift to remote teaching due to the COVID-19 lockdown. Adapting swiftly to this change, I utilized remote teaching tools such as online whiteboards and screen sharing to maintain an effective learning environment. Additionally, I remained highly accessible on Piazza, ensuring prompt responses to student inquiries. As a testament to these efforts, I received very positive feedback in the student evaluations at the conclusion of the Spring 2020 term:

"Guannan you are a real homie, thanks for carrying the course and helping out all the students who were lost as hell – please know that your help in this class is probably what got everyone through it, you are a blessed human being. Thanks for your existence :)"
"Guannan was patient and very helpful with the projects. I'm really glad he was my TA because he helped a lot along the way."
"Guannan is a great TA and is very talented. He was great at answering questions during the lab session and helping students debug their code, especially since most of us were using a language we had never used before. He was also very active on Piazza, helping me and other students whenever we had questions about the projects or lecture."

Lab Instructor, System Programming. I served twice as a Lab Instructor/TA for Purdue's undergraduate System Programming course. The course covers low-level C programming to create practical applications in the POSIX environment, including but not limited to, shells and web servers.

As part of my responsibilities, I supervised two weekly lab sessions, each with around 20 students. At the beginning of each new project, I prepared a comprehensive overview lecture to ensure students have a clear understanding to tackle the upcoming tasks effectively. This initial lecture served as a crucial point of reference for students, enabling them to grasp the core concepts and techniques necessary for their lab work. One challenge I encountered in this course is the considerable diversity of students, spanning from freshmen to juniors and encompassing both CS and non-CS majors in terms of their backgrounds. Through this experience, I recognized that effective teaching and communication can only be achieved by comprehending the diversity of student backgrounds and actively endeavoring to align the teaching objectives with their individual levels.

TEACHING INTERESTS

I would be excited to teach or develop any course about the design, implementation, and analysis of programming languages. Drawing from my past teaching experience at Purdue, I am well-prepared to deliver a modern *Compiler Construction* course, emphasizing a hands-on learning experience and optimization techniques for languages with high-level abstractions. I also have a keen interest in teaching courses on *Program Analysis and Verification*, focused on rigorous and formal techniques to reason about programs. Aligned with my research interests, I look forward to developing educational materials for emerging domains, such as compilation/reasoning techniques for smart contracts and quantum programs.

I am open to teaching a variety of programming courses (e.g. Functional Programming), with a specific emphasis on principled and pragmatic skills for thinking and reasoning about the construction of correct programs. Given my research experience in formal semantics and software engineering, I am also qualified to teach *Foundations of Programming Languages* (e.g., following the Software Foundations textbook), *Software Engineering*, and *Discrete Math*.

UNDERGRADUATE MENTORING

I was fortunate to mentor three talented undergraduate researchers [1] during my PhD and postdoc at Purdue. We met regularly, typically on a weekly basis, and collaboratively engaged in research projects that I led. My goal was to expand their understanding of how programming language research can be applied to solve practical problems while nurturing their technical skills and resilience in exploring solutions to research problems. Throughout this mentoring process, I have also developed communication skills in providing constructive feedback and effective guidance.

I started mentoring Shangyin Tan in June 2020 when he was a sophomore undergraduate student. In the project on constructing efficient engines for symbolic execution, Shangyin made significant contributions to several published papers (OOPSLA '20, ESEC/FSE '21, PEPM '22, ICSE '23). Notably, he is the first author on our PEPM '22 (co-located with POPL) paper. Due to his academic excellence, Shangyin received the *CRA Outstanding Undergraduate Researcher (Honorable Mention)* award in 2022. Currently, Shangyin is pursuing his research career as a PhD student at the University of California, Berkeley, under the supervision of Koushik Sen.

Yuxuan Chen worked with me on the staged abstract interpreter project (later published at OOPSLA '19) during his final semester (Fall 2018). Yuxuan subsequently joined Meta and is currently a senior software engineer, contributing to the development of the SPARTA static analyzer based on abstract interpretation. Mikail Khan began working with me on the development of compilation methods for concolic execution of WebAssembly in January 2023. He is planning to explore research opportunities in graduate school starting in Fall 2024. I am very proud that all my mentees have pursued PhD programs or engaged in industrial projects relevant to PL/SE research.

[1] I have obtained their agreement to discuss their experience in my statement.