TEACHING STATEMENT

Yahui Song
Research Fellow at National University of Singapore (NUS), Singapore
https://www.comp.nus.edu.sg/~yahuis/

My teaching philosophy is grounded in constructivism and positivism, which together create a dynamic and evidence-based learning environment. I believe that knowledge is constructed through active engagement and meaningful experiences while also being rooted in observable facts and empirical evidence. This dual approach allows me to foster critical thinking, problem-solving skills, and a deep understanding of the subject matter among my students. Education is a collaborative and transformative process. I strive to create an inclusive and engaging learning environment where students feel empowered to explore, question, and discover. Effective teaching goes beyond merely transmitting knowledge; it fosters critical thinking, creativity, and a lifelong love of learning.

1 Teaching Experiences

Throughout my graduate studies, I have dedicated over six hundred hours to teaching, engaging in various roles such as conducting tutorials, answering student inquiries, monitoring group activities, marking assignments and exam papers, and preparing supplementary materials for different modules. Conducting tutorials presented unique challenges, as it required me to provide clear explanations of assignment solutions and address the conceptual questions raised by students. I was honored to be recognized in the "Honor List of Student Tutors for Excellence in Teaching" for the academic year 2020-2021, reflecting my commitment to effective teaching.

I am deeply committed to teaching effectiveness and educational excellence, as evidenced by consistently high student evaluations from 2018 to 2021. For instance, in the CS3203 Software Engineering Project course, I received ratings of 4.3 and 4.8 in different semesters, surpassing the department averages of 4.1 and 4.2, respectively. Similarly, in the CS2104 Programming Language Concepts course, I achieved a rating of 4.4, above the department average of 4.2. While I received a rating of 3.7 for TIC2701 Principles of Programming Languages, which was slightly below the department average of 3.9, I view this as an opportunity for growth and improvement. Additionally, I have received valuable feedback from peers and students that highlights my ability to engage learners and foster a deep understanding of complex topics.

Furthermore, since 2020, I have actively engaged in conducting tutorials via Zoom and assisting with online exams using NUS Canvas. My responsibilities included setting up tests, managing online grading of student submissions, and providing support to students throughout their learning journey. Additionally, I served as a teaching assistant for the "Introduction to Computer Science" module at Yale-NUS, where I worked with a small class of 20 students. In this role, I developed test cases and provided individual feedback on code deductions from automated testing and code reviews each week through Canvas. This approach not only streamlined course delivery but also significantly improved student engagement.

Diversity in Teaching. Singapore is a highly diverse country, and the National University of Singapore (NUS) reflects this diversity with students from various ethnic, religious, and cultural backgrounds. Moreover, one notable area of diversity in computer science is the gender disparity. To promote student equity, I have advocated for the development of inclusive curricula that represent a wide range of perspectives and experiences. I collaborated with other lecturers to integrate more diverse voices and case studies into the curriculum, to enrich the learning experience for all students.

2 Developing an Undergraduate-level Course

In functional programming courses, students typically engage with a variety of classic exercises, such as implementing recursive functions like Fibonacci and factorial, utilizing functional programming operators like fold and map, and exploring monadic operators such as bind and return. They also experiment with different type systems. Customized exercises are tailored to specific languages, including list comprehensions in Haskell, higher-order imperative features in OCaml, and constraint-solving puzzles in Prolog. Beyond traditional topics, instructors can develop more innovative assignments that leverage advanced language features. For example, Scala, with its built-in support for asynchronous programming and concurrency, provides opportunities for such tasks. As a teaching assistant, I designed a hangman game assignment to help students practice concurrency operators in Scala, which received positive feedback from the students.

I have had the privilege of teaching a variety of programming-language-related courses during my graduate studies, including "Typed Functional Programming in Practice," "Principles of Programming Languages," and "Programming Language Concepts/Implementation." One of the most rewarding aspects of my teaching experience has been introducing students to declarative languages like Haskell and Prolog, as well as demonstrating imperative features in functional languages such as Scala and OCaml. These teaching experiences have sharpened my skills to manage time effectively, clarify complex concepts through concise examples, and stay abreast of the latest developments in programming languages, which have, in turn, inspired new research ideas.

3 Collaborating with Junior PhD Students

I have taken a leading role in the majority of my published conference papers, with a significant contribution to the remaining ones through refining ideas and enhancing the writing. Notably, I led our recent award-winning paper, overseeing the entire process from idea generation to implementation and final writing. My drive for high-quality research is fuelled by the excitement of discovery and the impact of contributing valuable insights to my field. Additionally, my aspiration to serve as a role model motivates me to continuously embrace new challenges and pursue further achievements.

I have also enjoyed collaborating with graduate students and cultivating an environment where ideas can flourish and be progressively implemented through teamwork. I include below a list of people that have enriched my research path:

1. I collaborated with Darius Foo, a Ph.D. student, on verifying higher-order imperative programs and algebraic effects. These two projects culminated in publications at APLAS22, FM24 and ICFP24.
2. I worked with Wenhua Li, also a Ph.D. student, on developing static analysis-based bug-finding tools for large-scale C and Java programs. This collaboration resulted in publications at FSE24 and APLAS23.
3. Additionally, I partnered with Yu Liu, a Ph.D. student, to create a symbolic program repair framework based on Datalog-defined analysis. Our paper is currently under submission.