January 14, 2025 (Get the Latest Version ☞)

TEACHING STATEMENT

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

I view teaching as a collaborative and mutually beneficial journey between students and instructors, where knowledge is built through active engagement and meaningful experiences grounded in observable facts and empirical evidence. Effective teaching transcends the mere transmission of information; it nurtures critical thinking, creativity, and a lifelong passion for learning. I strive to create an inclusive and engaging learning environment that empowers students to explore, question, and discover. My primary passion lies in programming and programming languages. Throughout my teaching journey, I aim to inspire students to appreciate complex concepts while developing their practical problem-solving skills for effective computing applications. Moreover, I am committed to mentoring students at all levels, guiding them as they pursue their aspirations of becoming future researchers or engineers in both academia and industry.

1 Teaching Experiences

Throughout my graduate studies, I dedicated over six hundred hours to teaching, engaging in various roles, including conducting tutorials, monitoring group activities, grading assignments and exam papers, and preparing supplementary materials for different courses. Each of these responsibilities presented unique challenges and required diverse experiences, which I approached with patience and determination. My commitment to teaching effectiveness and educational excellence is reflected in the consistently high student evaluations I received from 2018 to 2021. This dedication culminated in my recognition on the **Honor List of Student Tutors for Excellence in Teaching** for the academic year 2020-2021. I am proud of this achievement, as it underscores my passion for fostering a positive learning environment and supporting student success. Serving as a teaching assistant (TA), there are several experiences stand out as particularly impressive:

- The CS3203 Software Engineering Project course provides students with hands-on experience working in project groups through a complete software development lifecycle to develop a well-designed, well-tested, large-scale software system. Students learn about agile practices such as iterative development, continuous integration, and regular feedback loops to enhance project outcomes. I was responsible for monitoring the progress of multiple groups and providing guidance when challenges arose related to teamwork communication, algorithm design and timetable scheduling. Given the large scale of the course, I focused on ensuring that each student received the attention they needed to succeed.
- The CS2104 Programming Language Concepts course introduces foundational concepts for numerous programming languages, covering paradigms such as imperative, object-oriented, functional, logic, constraints, and concurrent programming. These concepts are illustrated through examples from languages like C, Java, Haskell, Scala, and Prolog. I thoroughly enjoyed my TA experience in this course, which broadened my perspective on programming languages. My responsibilities included supervising weekly tutorial sessions and assisting students with complex topics such as pseudo-code interpretation, static and dynamic semantics, and type inference. At the start of each new project, I prepared a comprehensive overview lecture to ensure that students had a clear understanding of the upcoming tasks. This role not only deepened my grasp of these concepts but also enhanced my ability to explain intricate ideas clearly and effectively.
- The YSC1212 Introduction to Computer Science course at Yale-NUS provides a comprehensive introduction to essential computer science concepts for both CS and non-CS majors. It emphasizes the development of algorithmic thinking and foundational knowledge suitable for students from diverse backgrounds. Primarily taught in OCaml, this course encourages practical engagement with functional programming concepts within

a small class of 20 students, fostering personalized interaction. As a teaching assistant, I developed thorough test cases and provided weekly individual feedback on code through automated testing and code reviews via Canvas. This approach streamlined course delivery and significantly enhanced student engagement, ensuring that each student received the necessary support to succeed in their programming endeavours.

Since 2020, I have actively engaged in conducting tutorials via Zoom and assisting with online exams using NUS Canvas 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. One of the significant challenges I encountered was the considerable diversity among students, ranging from freshmen to juniors and including both CS and non-CS majors. This experience taught me that effective teaching and communication depend on understanding students' diverse backgrounds and aligning teaching objectives with their individual levels. Recognizing and connecting with this diversity is essential for creating an inclusive educational environment, as it fosters engagement and enhances learning outcomes for all students.

2 Teaching Interests

I would be excited to teach or develop a programming language design, implementation, and analysis course. Drawing from my past teaching experience at NUS, I am well-prepared to deliver a functional programming course that emphasizes hands-on learning experiences and optimization techniques for languages with high-level abstractions. Aligned with my research interests, I also have a keen interest in teaching courses on Program Analysis and Verification, focusing on rigorous and formal techniques to reason about programs. This could include topics such as program semantics and verification methods that ensure program correctness and reliability.

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 Graduate Student Researchers

I have thoroughly enjoyed collaborating with graduate students and fostering an environment where ideas can flourish and be progressively implemented through teamwork. Here are the talented graduate student researchers I had the pleasure of working with during my PhD and postdoc at NUS:

I collaborated with Darius Foo, a PhD student, to tackle non-trivial verification tasks in functional programming through specification design, separation logic, temporal logic, and inductive proofs. Our work on verifying higher-order imperative programs and algebraic effects has resulted in publications at APLAS22, FM24, and ICFP24. We are excited to continue our collaboration and further explore these complex research areas. I worked with Wenhua Li, also a PhD student. Our mutual interest in bug-finding resulted in the development of static analysis-based tools for large-scale C and Java programs (imperative OOP). This collaboration produced publications at FSE24, APLAS23, and TACAS25, which propose practical tools for specification design and inference. I partnered with Yu Liu, another PhD student, and our shared interest in developing static analysis-based repair techniques has fostered regular discussions. Together, we created a symbolic program repair framework based on Datalog-defined analysis, with our paper currently under submission. I supervised Yan Dong's thesis, a Master's student who developed an SQL frontend on top of OCaml for data analysis. The thesis was presented at IFL22.