

Teaching Statement

Hou Ruomu

Nov 2024

My Teaching Journey

Teaching has always been a source of joy and fulfillment for me, and my journey as an educator began naturally. In high school, I embraced opportunities to deliver short lectures on diverse topics. During my undergraduate years, despite the demanding workload of a double-degree program, I actively sought teaching roles. I served as a teaching assistant (TA) for the course CS3230: Design and Analysis of Algorithms three times. My responsibilities included conducting weekly tutorial classes (~20 students), preparing and grading assignments, and holding weekly office hours.

As a Ph.D. student, I expanded my teaching portfolio, taking on TA roles in several other courses. Beyond formal teaching, I volunteered as a mentor in my university residence, offering academic guidance in computer science and mathematics. I also extended my mentorship online, helping an international student gain admission to our CS department and subsequently supporting his academic and career growth. We continue to meet and exchange ideas regularly. These experiences have deepened my passion for teaching and reinforced my commitment to student development.

During my 10 years at the university, I witnessed many extraordinary developments in computer science education: a sudden increase in student intake, with students having varying levels of interest in the subject and differing motivations—ranging from genuine passion to securing lucrative graduate salaries. The number of teaching assistants for a single course grew from 8 to 80 in just a few years. The rapid adoption of large language models (LLMs) and online resources transformed the learning landscape.

These changes present significant challenges for educators: i) Managing an ever-increasing cohort size; ii) Addressing vast differences in students' prior knowledge, levels of interest, and personal goals. iii) Adapting to the fast-evolving knowledge and tools in the field, such as ChatGPT.

As a CS educator, I believe education must stay attuned to new trends and the evolving needs of students to impart appropriate knowledge and foster a meaningful, fulfilling learning journey for every individual.

Teaching Fundamentals

In education, certain principles remain constant, providing a foundation for effective teaching despite the ever-changing environment. These fundamentals guide my teaching philosophy:

1. **Care for students.** While it is gratifying to nurture talented students, the heart of teaching lies in supporting those who face challenges. Through my experiences as a TA and mentor, I have observed that fear of failure and lack of confidence often hinder students' learning. A key part of my teaching approach is addressing these barriers through empathy and encouragement. For instance, as a senior undergraduate peer mentor, I worked with a first-year student struggling with functional programming. His difficulties led him to question his decision to study computer science. By relating functional programming concepts to his prior experience in programming and dedicating time to clarify the basics, I helped him rebuild his confidence. He went on to graduate with first-class honours, a testament to the transformative power of personalized guidance.
2. **Availability of Learning Opportunity.** Students learn best when they have access to ample resources for practice. For instance, while assisting in *CS1101S: Programming Methodology*, I developed a web-based programmable pattern-manipulation platform. It allows us to create many tasks and primitives for student to play with. This tool provided a variety of opportunities, enabling students to practice programming concepts as much as they desire to, until they gained confidence.
3. **Continuity in Learning.** Revisiting previously learned material reinforces understanding and

promotes deeper insights. I often design tutorials and assignments that connect new topics with earlier material. For example, in algorithm courses, I reuse versatile tasks—like word counting and prediction problems—to explore multiple algorithmic approaches.

4. **Catering to Different Learning Speeds.** Teaching should cater to a range of student abilities. An effective strategy is to design assignments with varying levels of difficulty. For example, a course I attended allowed students to complete weekly quizzes by solving either a few complex problems or many simpler ones. This flexible approach was well-received by students of all levels.

Adapt Education to The Evolving Field

As the field of computer science evolves rapidly, educators face new challenges. I continuously explore innovative techniques to address these changes. Two examples stand out:

1. **LLM Models.** The advent of large language models (LLMs) like ChatGPT has revolutionized education. Students often find that LLMs provide clearer explanations than traditional lectures. I, too, leverage LLMs for learning and recognize their potential to deliver personalized, round-the-clock support. As educators, we must embrace this shift and adapt our teaching strategies. For instance, programming courses should place less emphasis on generating compile-able and correct code—since students can easily use LLMs for that—and instead focus on logical clarity, pseudocode development, and debugging skills. Additionally, identifying corner-case bugs, a known weakness of LLMs, should receive more emphasis in the curriculum.

By designing LLM-friendly course materials and rethinking our roles, we can ensure that students gain the skills to learn effectively in an AI-enhanced academic environment.

2. **Larger Class Size.** When I took the introductory course *CS1101S* in 2014, there were ~50 students. By 2021, when I served as a TA for the course, enrollment had grown to over 500 students. The learning experience changed significantly. In 2014, the lecturer knew every student's name, and the classroom environment was interactive. In 2021, lectures had to become more one-directional out of necessity. To preserve the quality of education, the teaching team (including myself) embraced technology and implemented several solutions:

- **Responsive Online Forums.** We created an online forum where TAs were on duty in rotation to provide timely responses to student questions.

- **Gamified Group Learning.** Each TA led a group of students in a gamified system to foster teamwork and support.

Looking ahead, I envision applying cutting-edge technologies—such as live broadcasting tools with interactive features like Danmaku—to bring interactivity back to large classrooms.

Courses that I can teach

Based on my academic background and interests, I am well-suited to teach the following courses:

1. **Algorithm Design and Analysis**
2. **Introduction to Computer Programming**
3. **Introduction to Computer Networks**
4. **Computer Security / Network Security**
5. **Cryptography Practices**
6. **Distributed Algorithms and Systems**
7. **Blockchain Systems**
8. **Randomized Algorithms**