

Hao Wang | TEACHING STATEMENT

TEACHING PHILOSOPHY AND EXPERIENCE

Both of my parents are teachers. Their persistent passion and dedication for teaching have planted a seed in my young mind that I would also be a teacher one day. During my study and research, I have benefited a lot from my passionate and engaging teachers. Their encouragement and support further strengthen my belief in becoming a teacher. However, only enthusiasm in teaching does not suffice to make a good teacher. Four years ago, when I gave my first tutorial to students of the course “CSC458 Computer Networks”, I received unexpected silence from my students, even though I had prepared my slides carefully. I realized that students still might not understand those well-prepared explanations and examples, which seemed crystal clear to me.

Over the next four years, I have been improving my teaching practices. I joined “APS1203 Teaching Engineering in Higher Education” this year, a course designed for senior Ph.D. students to cultivate their understanding of engineering pedagogy. My teaching strategies have been evolving from a content-centric approach to be more student-centric. My teaching philosophy is now grounded in the belief that it is my responsibility to foster an engaging and interactive learning environment for students. And I believe creating and maintaining this environment requires the following elements:

Motivating Students

Motivating students' interests is a top priority in my teaching philosophy. I engage my students by providing an interactive and rewarding learning experience. When I was a teaching assistant of the course “ECE454 Computer Systems Programming”, another teaching assistant and I developed an online testing platform that automatically evaluates the performance of code submitted by students and instantly returns the results to them. Students can figure out how to improve their code with the hints from the testing platform. Besides, we added a ranking feature that evaluates submissions and exhibits the top ten performance in real-time. Inspired by the rank, students endeavoured to learn the computer system internals and keep enhancing their code. From the reviews collected from the course evaluation, I found students enjoyed the instant feedback provided by the online testing platform.

Valuing Feedback

I believe teaching is an interaction between a teacher and students with different background and prior knowledge. Feedback from students helps me improve my teaching strategies and materials. I proactively collect feedback by making small talks with students during breaks. For example, before I gave tutorials to students of the course “CSC369 Operating Systems”, I usually asked them questions such as how hard they thought the assignment was and any problems they had encountered. They once complained about the cumbersome process of debugging on a remote server via the SSH protocol. I recommended that they could use the editor Visual Studio Code with its remote deployment plugin. Students applauded for my demonstration of how to debug with the editor remotely. Without feedback from students, a teacher could hardly prepare for potential problems that students might encounter. Therefore, as a teacher, I am responsible for encouraging students to share both their positive and negative learning experiences.

Building Connections

Connecting course materials to students' previous knowledge and real-world examples consolidates their understanding of the new concepts. I have been serving as a teaching assistant of the course “ECE353 Systems Software” for three years, one of the most frequently asked questions was how to understand conditional variables, which is one type of locking mechanism in operating systems. To answer this question, I first review the concept of semaphores, which is

a similar type of lock mechanism. Then, by comparing the two mechanisms with a producer-consumer example, students found it easy to develop a relevance that facilitates their understanding of conditional variables. I also applied this teaching strategy in the course “ECE444 Software Engineering”. In this course, students were grouped into teams and collaborated on a project with the code management system, Git. At first, they were confused by all the sophisticated Git commands, wondering why they cannot copy and paste their code. I explained the necessity of Git by sharing my internship experience that how Git efficiently simplified project management and quality control at Intel. Students then realized that following a strict Git workflow will save them a lot of time, even for their small team.

Encouraging Problem-Based Learning

Ultimately, I am committed to promoting students’ self-motivation and independent thinking. I have been actively practicing problem-based learning, a teaching strategy that students learn a subject by solving open-ended problems. I have mentored nine undergraduate students with their thesis projects and summer projects, two of which are women students. One of them is Zakhary Kaplan, who worked with me on my ongoing project of a new machine learning system this summer. Back then, Zakhary was a junior student without any knowledge of machine learning yet. When I introduced my progress to him, he hesitated about what he could contribute to the project. He was panic when new concepts kept emerging. I split an unsolved problem of my machine learning system into a series of subproblems and assigned them to Zakhary in a sequence from easy to difficult. By solving those problems, Zakhary learned machine learning quickly and built up his confidence that he would contribute more to the project. By the end of this summer, Zakhary and I summarized our work into a paper, which has been accepted by IEEE INFOCOM 2020.

TEACHING INTERESTS

I have served as a teaching assistant in seven different courses so far, including operating systems, software engineering, database systems, and computer networking. My teaching interests span a variety of topics and subjects in computer science and engineering. I can offer courses at both undergraduate and graduate levels.

For undergraduate-level courses, I am interested in teaching courses at multiple levels: Programming Fundamentals (C/C++, Python, Java, and Scala), Operating Systems, Software Engineering, Computer Networking, and Database Systems. For graduate-level courses, I can teach in the following courses: Introduction to Cloud Computing, Distributed and Parallel Systems, Large-Scale Data Processing, Software Engineering, Network Architecture and Protocol, and Inference Algorithms and Machine Learning.

These courses are related to my expertise and research areas. I will incorporate the latest research into my teaching to improve students’ hands-on ability and inspire their critical thinking. Also, I would like to open a few research projects for undergraduate students. Undergraduate students will have a chance to experience the excitement of solving research problems.