Teaching Statement

Rong Ge

I enjoy teaching and interacting with students and novice researchers. This is the reason I chose the academic career. Today, I feel fortunate that I am still in academia and have the same passion.

As an instructor in computer science, I am focused on several equally important objectives in designing and teaching courses. First and foremost is to disseminate and pass the fundamental core knowledge and problem-solving skills to students. Second, I strive to keep students challenged at the right levels so that students maintain their confidence while reaching their potentials. Third, computer science and information technology consistently and rapidly evolves and applications emerge from many scientific, engineering, social, and financial domains, and I strongly believe the importance of my role in exposing students to new exciting technology and applications, and to keep students up-to-date with current technology.

I use a variety of strategies to accomplish these objectives inside and outside of classrooms, which includes but are not limited to the following.

Effectively disseminate fundamental and core knowledge and problem-solving skills. To this end, I strive to keep students interested and confident, and get them maximally engaged during class and through assignments. Interest is the best teacher and being confident in one’s abilities is requisite for a student to learn effectively. I carefully design course materials to keep students interested and motivated. When I teach data structures and algorithms course, I start with with applications to real world problems. For example, I provoke students to think of graph applications when they first encounter graph traversal algorithms. With little hints students are able to relate graphs to maps in GPS, transportation traffic control, internet routing, and social networks that students use in everyday life. I often use real interview algorithm questions from Google, Facebook, Microsoft, and other companies when introducing algorithm design techniques. These questions are fun, representative, and used in real-world computing services and software.

Active learning takes place when students are engaged in listening, seeing, discussing, doing and thinking during class. Computing concepts are abstract and could be hard for students to grasp. I use tools and extensive visual techniques such as graphs and animations on both blackboard and powerpoint slides to illustrate how algorithms work step by step. For example, I bring poker cards to data structures and algorithms classrooms and let students use the cards to mimic sorting algorithms. I design lectures to be interactive, both with me and with peers in groups. Students rated my course with nearly perfect score for the question of whether the instructor encouraged student participation in course evaluations. I watch students’ reactions and adjust my pace accordingly, and endeavor to create welcoming classroom environments for interaction, especially when students hesitate to ask questions. Students have been giving positive feedbacks for these efforts.

Keep students challenged. Gaining the fundamental and core knowledge and skills could mean challenges, particularly when high standards are maintained. I carefully gauge and assess students’ ability, design materials and adapt to meet students ability. For example, algorithm course requires strong math skills in analyzing algorithms’ running times and solving complex combinatorial problems. Our students often have forgotten those math skills. I introduce and review relevant math contents, and assign math homework problems to reinforce the understanding so that students are better prepared for algorithm designs. All the courses
that I have taught, including operating systems, computer architectures, and compilers, are constantly considered more challenging and yet valuable and meaningful than the same-level courses, according to student course reviews. To make sure students are able to face the challenges, I incorporate detailed background materials into lecture notes and post them in advance, offer timely responses and discussions, post online sources and textbooks for topics, practice important concepts in classes, and carefully design assignments.

*Keep students up-to-date.* “Rowing a boat against the current; if it does not move forward it will move backward.” Computer/computing science and technology are developing rapidly in every dimension including hardware, software, and application. I seek to incorporate the latest technologies into both undergraduate and graduate course materials to keep students with up-to-date technologies and future trends. I investigate data structures and algorithms that popular employers such as Google, Facebook, Microsoft use in their systems and services and integrate them into algorithm courses. I also introduce the cutting-edge technology including machine learning workloads, accelerated computing with GPUs and FPGAs and ASICs, and Big Data storage and processing technology into relevant courses.

I frequently **assess my teaching effectiveness using multiple resources** including students, colleagues, and self-reflection. Students have been an important resource for my self assessment. I collect anonymous feedbacks from students using questionnaires in classrooms, and informally ask their feedbacks during casual conversations outside of classrooms. I also use students’ inquiries, feedbacks from casual conversations, and surveys to gauge my teaching effectiveness and students’ learning. My colleagues and friends have been providing constructive and valuable advices and feedbacks. I have learnt a great deal from them on every step and in every year during the course of my teaching career, including knowing students and inspiring them, organizing classes, engaging students in class, etc. Self-reflection is a regular exercise that I practice right after coming back from classes, finishing grading assignments and exams. I reflect on what has gone well and not so well as expected, and note improvements that I can make next time.

I enjoy teaching a variety of courses ranging from data structures and algorithms to systems and architectures for both undergraduate and graduate students. “To teach is to learn twice over”.