Philosophy of Teaching – Yuanyuan Jiang

In an ever-changing field like computer science, the knowledge gained from college is only a gateway to a journey of life-long learning. I see my role as a “teacher” being a facilitator and helper for students' learning. I strive to create a learning environment that supports my students to gain passion for the field, engage in problem solving, collaborate with their peers, and obtain an interdisciplinary perspective on computing.

Student-Centered Teaching

A common challenge in college teaching is that students have different backgrounds and learning abilities. My class is tailored to provide support for diverse learners, from ones who found a subject challenging, to ones who are experienced with the topics. When designing the courses, besides the common core learning objectives, I provide multiple paths and strategies for students to choose to finish the course requirements. Throughout the class, I encourage students to focus on learning a useful skill by providing self-reported goal-establishing and evaluating surveys, by focusing on learning objectives instead of scores, and by celebrating everyone's achievements, big or small, with the same enthusiasm. It was very encouraging to see my students take risks and pick assignments that promoted their own learning and heightened their sense of accomplishment.

As a caring and supportive person, I might not be the most humorous lecturer, but I let students know that I care about their growth and support them as individuals. Students are more cooperative and engaging when they are known by the teacher and treated with understanding and hope. I vividly remember two students in my introduction to programming class. One of them quickly finished all the required course. I remember the excitement and drive in his eyes when I gave him hints on where he could search for answers by himself for extra project. The other student faced more challenges. When I saw him working on past-due homework problems, I tutored him on the problems and celebrated his success just like he was doing new practice, without questioning why he was behind on homework. The day when I helped him debugging and got his first programming project running, he burst into tears in the lab and overwhelmed by how far he had gone through. These are the moments that made all my efforts worthwhile.

Creating Collaborative and Engaging Learning Environment

In the Bloom’s Taxonomy learning pyramid, applying and analyzing has higher cognitive complexity than remembering and understanding. This is where I spend a large part of the in-classroom time with active learning activities. Instead of hearing content lecturing, students actively participated in designed activities like hands-on problem solving, group discussion, peer coding, etc. Instead letting students write down and remember what I put up on board, I typically set time after hands-on sessions and ask students to discuss and share what they’ve learned and form a class-wide shared notes on site by themselves. They get timely feedback and guidance in the classroom to navigate through the more challenging parts in college learning: applying what you read to solve new problems, and are more engaged and gain better learning outcomes in this active learning process.

Collaboration is an essential yet hard to obtain skill in computer science. I believe the collaboration skill need to be embedded into different stages of training which will promote student learning even in traditionally solo learning subjects. I try to create a collaborative learning environment by establishing a non-competitive course requirement and grading strategy, providing coding partners, encouraging online forum discussion, etc. It was common to see my students meet with their coding partners outside of class, and that some of the fast learners naturally play a teaching assistant role. With a sense of community, students are more engaged and eventually learn how to find good study-partners, to communicate technically, and work as a team in the process.

Cultivate Skills
Besides learning materials, students need to learn skills in college to become professional with specialties. I embed skill training in my teaching. Problem solving is the centered skill and the fun part of computer science. When introducing new concepts, I always challenge the class using real-world problems, let the students discuss easy solutions, and then introduce different scenarios or complications to give them a chance to find the complete solution by themselves and experience a sense of accomplishment. In office hours, instead of directly solving students’ problem one by one, I let all of them into the room and participate group discussion as an equal partner to show them how to reach the solution together. In project designing, I incorporate interesting background, like linking programming practice to encryption and a Sherlock Holmes novel, to let students solve interesting problems even in introductory courses. The problem-solving interest and skill cultured through a semester will extend to exams as well. I believe even exams should be part of learning with a focus on problem solving, not on repetition of memorized content.

As computer science is getting more influential in other disciplines, students who have an interdisciplinary view can gain an edge on being innovative and solving exciting interdisciplinary problems. As an interdisciplinary scholar myself, I worked with psychologists, engineers, and artists. I saw how we collaboratively solve challenging research problems and bring it to the classroom. For example, I routinely show students exciting research videos like how physics is used in animation production and how computer graphics are used in shadow theater. Open major students also present what they found on computer science impacting other disciplines.

Continuous reflection and improvement

I believe good teaching is not a born talent but a skill that can be learned, trained, and improved. Teaching tips and mimicking good teachers experienced as a student is just a starting point. Though not a common choice in my department, I enrolled in the Certificate in College Teaching program. I took classes from the College of Education department, read literature on college teaching, worked as a teaching fellow for an educational psychology class. Cognitive psychology is part of my personal and research interest and I closely collaborated with developmental psychologists in research. These experience and interests open a window for me to see computer science education and my classroom from a different perspective. Like other types of research, pedagogical theories can be a powerful tool for guiding teaching practice. For example, when designing courses, instead of solely focusing on knowledge delivery, I take student motivation and learning objectives into consideration. When facing problems, I can ask for help from experienced educators in and outside the major to get a more rounded view.

A well-received class can still be improved to better fit a certain student group and improve learning outcomes. I taught the Foundations of Computer Science course for open majors twice at Cornell College. I fully designed the course structure and material by myself. Students performed great with majority finished projects with clean programming organized into function and modules. I received very positive student reviews including comments such as: "My proficiency in this class clearly evolved with each exercise, project, and exam that we took", "Yuan was a visiting instructor, but I really enjoyed her class. She was available and accessible. She created 3 projects for use that ranged from practical like writing multiple data files to a single Excel CSV file to creation of a remedial Minesweeper game. Yuan adapted well to the OCAAT system and I would like to see her teach more courses here at Cornell. She made sure to keep students with both previous experience in CS and without engaged." Despite the success, I worked to improve the course and saw clear progress in the second block. More students tried/finished the optional challenge project (5 out of 12 vs. 2.5 out of 14). Though the course covered more content with harder homework and exams, student scored higher and reported less perceived material difficulties and time spent outside of classroom.

Overall, I view college teaching as an iterative process of constant reflection and improving and a way to share my passion to the computer science field to my students. I strive to provide an engaging and enjoyable learning experience for my students that not only lets them learn the content, but also gradually gain interests and skills to become a problem solver and life-long learner in this fast changing world.