Teaching Statement – Yuanyuan Jiang

In an ever-changing field like computer science, the knowledge gained from college is a gateway to a lifelong learning journey. 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 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 classes are tailored to support diverse learners, from ones who find a subject challenging, to those who are experienced with the topics. Besides the common core learning objectives, I design classes with multiple paths and strategies for students to choose from. Throughout a class, I encourage students to learn useful skills 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 taking challenging 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 the teacher knows them and treats them with understanding and hope. I vividly remember two students in my introduction to programming class. One of them quickly finished all the required assignments. I remember the excitement and drive in his eyes when I gave him hints on where he could search for answers by himself for the extra-credit 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, without questioning his progress. The day when I helped him debugging and got his first programming project running, he burst into tears in the lab, overwhelmed by the progress he had made. These moments 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. Therefore, I focus on active learning activities in classroom to engage students. Students actively participated in designed activities like hands-on problem solving, group discussion, peer coding, etc. Instead of providing notes, I encourage students to share what they've learned and form class-wide shared notes among 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 collaboration skills should 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 share their experiences. 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, for example, is a critical skill and the fun part of
computer science. When introducing new concepts, I always challenge the class with real-world problems, let the students discuss easy solutions, and then introduce different scenarios or complications. Students have a chance to address the problem by themselves and experience a sense of accomplishment. In office hours, instead of guiding students individually, I participate in group discussions with them as an equal peer to show them how to reach the solution. In project designing, I incorporate interesting background stories, like connecting 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 cultivated 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 becoming more influential in other disciplines, students who have an interdisciplinary view can gain an edge on innovation. As an interdisciplinary scholar myself, I worked with psychologists, engineers, and artists. I experienced how we can collaboratively solve challenging research problems and I brought 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 were also encouraged to present what they found on computer science impacting other disciplines.

**Continuous reflection and improvement**

I believe good teaching is not an innate talent but a skill that can be trained, and improved. Teaching tips and learning from good teachers are just the starting points. 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, read literature on college teaching, and worked as a teaching fellow for an educational psychology class. These experiences opened 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 of the major to get a more well-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. Most students finished projects with modular programs organized into functions 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." Based on the success of my first class, I continued to improve the course and saw clear progress in the second block. More students 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 scores higher and reported fewer difficulties and time spent outside of classroom.

Overall, I view college teaching as an iterative process of constant reflection and improvement, and a way to share my passion for computer science with 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.