Teaching Statement

Xiang (Jenny) Ren

During my graduate training, I taught tutorials and conducted lab sessions for introductory programming, operating systems and parallel programming classes. I also mentored several master's and undergraduate students. This experience helped me develop confidence and interest in teaching and mentoring.

My goal in teaching and mentoring is to facilitate positive transformations. As a teacher, I strive to help students gain skills and knowledge to become skilled computer science professionals. As a mentor, I strive to guide students to become independent thinkers and productive researchers. I find watching such transformations take place meaningful and rewarding, which draws me to the career of an educator.

Teaching experience As a graduate student at the University of Toronto, I have served as a teaching assistant for operating systems, introductory programming classes in C and C++, as well as parallel programming at the graduate level. My responsibilities include developing tutorial content and teaching tutorials, helping students with their programming assignments during lab sessions, and grading their assignments and exams. I conducted lab sessions for APS105-Computer Fundamentals and received a 6.7/7 from anonymous teaching evaluations released by the course coordinator. Students commented that I was helpful and patient and effectively taught them important programming skills like debugging. I taught weekly tutorials for ECE244 Programming Fundamentals and my tutorial session had much a higher attendance than the average.

Teaching philosophy My approach to teaching is to keep the students engaged and to promote active learning. The prerequisite to student engagement is clarity in communication and presentation. To achieve this, I thoroughly prepared for tutorial or lab sessions to break down complex concepts into intuitive explanations for the students. For teaching algorithms such as linkedlist traversal, quick-sort, etc., I created animations to help students visualize how each step of the algorithm works. In addition, I noticed that if students become confused at some point during a tutorial session, it can be hard for them to follow through with the rest of the materials. So, I tried to gather feedback from the students as I taught in class, especially after teaching harder-to-grasp concepts. Based on students' feedback, I had chosen to clarify in class or review in the next class.

I also tried to teach interactively during my lab or tutorial sessions. During lab sessions, students often struggled with the same bugs, but directly telling them the cause of the bugs did not help improve their debugging skills. Instead, I encouraged my students to put on the "detective hat" while I diagnosed the bug with them and guided them toward the root cause logically. Such exercises helped the students gain confidence in independently carrying out the debugging process. For my tutorial sessions, I tried to get the students to participate in class by breaking down the materials into puzzle-like exercises with incremental difficulty and encouraging the students to answer. The interactive process was fun and energizing for both the students and me.

To keep the students engaged, it was also crucial for the students to care about what they were learning. Not every student has an inherent interest in the subject. Whenever possible, I explained to my students why a particular piece of knowledge is relevant to the real world. For example, when I taught the tutorials for the introductory C++ course, I told my students that popular software systems like Chrome, MySql, etc., were written in C++. I also explained that even though C++ had constructs like pointers and references, which could be harder to grasp, having good knowledge of the language would give them an edge in building system software and allow them to easily pick up other object-oriented languages like C# or Java.

Teaching interests I have teaching and research experience in a variety of computer science topics, including system programming, operating systems, distributed systems, compilers, program analysis, software engineering, and more specialized topics such as system performance analysis and system reliability. I'm interested in teaching and or developing both foundational undergraduate courses and more advanced graduate classes.

Mentoring approach A major reason I am excited to work as a professor is the privilege of working with students. I have so far mentored three master's students and eleven undergraduate students. From working with these students, I learned it's important to first understand of the students' skill levels, as well as work and communication styles, then adapt to them accordingly. To keep inexperienced junior students motivated, I tried to set reachable goals for them, and offer appropriate notice when they were stuck while allowing them to try things out first and challenge themselves. As the student progresses in skills, knowledge, and research "taste", so should their level of independence. I believe the goal of graduate school training is to produce skilled researchers and independent thinkers. As a result, I've always encouraged junior students I've worked with to think deeply and creatively, which I believe is what leads to great research. For example, while reading papers to learn about state-or-art research is important, I encouraged junior students to trust their intuitions and think outside of the box of previous research. Finally, a good mentor-mentee relationship rests upon a solid interpersonal foundation. My goal is to create a supportive environment to help students grow and find their success.

A Teaching Experience at the University of Toronto

- **Teaching Assistant**, ECE 1747H Parallel Programming   2020
  Graded and assisted students with final projects.

- **Teaching Assistant**, ECE297 Design and Communication   2019
  Graded and assisted students with programming assignments.

- **Teaching Assistant**, ECE 244 Programming Fundamentals   2015, 2016, 2018
  Taught and developed materials for weekly tutorial sessions.
  Assisted students with programming assignments.

- **Teaching Assistant**, CSC369 Operating Systems   2018
  Graded and assisted students with programming assignments.

- **Teaching Assistant**, APS105 Computer Fundamentals   2017
  Graded and assisted students with programming assignments. **Student rating: 6.7/7**

- **Teaching Assistant**, ECE344 Operating Systems   2016
  Graded and assisted students with programming assignments.