Statement of Teaching

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, and 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 included 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 [1]. 75% of the students who participated in the evaluation rated me as "outstanding." 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 higher attendance than the average.

Teaching philosophy I believe learning should be a hobby and not a chore for the students. To keep my students engaged, I prioritized to communicating and presenting course materials clearly. I prepared for tutorial or lab sessions thoroughly to break down complex concepts into intuitive explanations for the students. To teach algorithms such as linkedlist traversal, quick-sort, etc., I created animations to help students visualize how each step of the algorithm worked. 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. To avoid this, 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 would clarify in class or review in the next class.

I found keeping my lab or tutorial sessions interactive and fun an effective way to encourage the students to think and learn actively. When students encountered difficult bugs during lab sessions, I encouraged them to put on the "detective hat" while I diagnosed the bugs with them and guided them toward the root causes logically instead of directly telling them the root causes. Such exercises helped the students gain confidence in independently carrying out the debugging process. For my tutorial sessions, I broke down the materials into puzzle-like exercises with incremental difficulty. I encouraged each student to think about the exercises and come up with their own answers. Teaching interactively and incorporating fun elements boosted the energy of the class and increased enthusiasm for learning.

To motivate the students' interest in learning, it is also crucial for the students to care about the course material. 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 systems 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, parallel programming, compilers, program analysis, software engineering, and more specialized topics such as system performance analysis and system reliability. I'm interested in teaching and developing both foundational undergraduate courses as well as 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 to understand and adapt to their strengths and weaknesses, as well as collaboration and communication styles. I also tried to be considerate of what types of projects or tasks interested the students the most or benefited their careers the most. To keep inexperienced junior students motivated, I tried to set well-defined, reachable goals and offer appropriate help when they were stuck while allowing them to try things out first and challenge themselves. As the students progressed in skills, knowledge, and research "taste," I encouraged them to work more independently, to trust their intuition, and to think outside of the box of previous research. I believe the end goal of graduate training is to produce independent researchers who are deep and creative thinkers, and the process of graduate training should be one where open communication is encouraged, and each student is supported and cared for by the advisor.

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.

References

[1] 2017.   APS105 Teaching Assistant Evaluation.   `https://jrenx.github.io/cv/APS105_student_evaluation.pdf`. (2017).