Teaching Statement

Xu Zhao

Practice makes perfect, and teaching is no exception. Throughout my time in graduate school, I have worked as a TA for several computer system courses, and even had the opportunity to instruct a systems course this past Fall (see Appendix I). These courses cover a broad range of topics in the computer systems field, including operating systems, computer systems programming, and compilers. I believe that teaching a systems class is about more than just conveying concepts in the classroom; it is also about inspiring solutions to real world problems that occur in the practice of engineering. Therefore, my work as a TA, which included redesigning lab materials and conducting in-lab help sessions, emphasized both general design principles and hands-on coding experience.

Beyond my work as a TA, I took great pleasure in being the sole course instructor for **CSC467: Compilers and Interpreters**. This allowed me to redesign the curriculum, rather than reusing the slides and lecture materials from prior lecturers. (I reused the lab materials from the prior lecturer, Prof. Jianwen Zhu.) In particular, I put a lot of effort into improving the practicality of the course material. For example, I introduced the topics of compiler design in object-oriented and functional programming languages. All these teaching experiences have motivated me to seek an academic career with the following teaching philosophy.

Motivation. Provoking and developing students' interest in the course material is a critical part of teaching, a part that I have thought about carefully. A few examples from CSC467 serve to illustrate my approach: to explain the structure of a stack frame, I conducted a live demo of an x86 stack overflow exploit gaining shell access; to demonstrate the optimization of unstable code, I used another live demo showing that the same code generates different results under different optimization levels. Such practices stimulate students' interest and improve their engagement throughout the class. Generally, students get excited about problems that are both intellectually stimulating and relevant to real world scenarios.

Respect. I respect and give great care to every student in my class. For example: I usually leave a few minutes at the end of every class for students' questions, and I answer each question patiently until everyone is crystal clear about the material; I contributed to 232 answers on the Piazza forum (81% of the total instructors' contributions); I developed and released an anonymous feedback system on the course website before the very first class such that I could react to students' feedback at anytime[1]. This hard work has paid off, as can be seen from the students feedback (see Appendix II). In particular, one student gave me a special suggestion to teach an advanced compiler course in the following semester.

Expertise. I demonstrate my expertise on the subject matter throughout the class. First, I prepare each lecture very carefully, making sure I clarify any subtlety, and getting ready for any questions. As evidence, I have received students' feedback characterizing me as "*very knowledgeable of the relevant topics*" and able to "*breeze through complex topics*" through-

---

[1]The evaluation of CSC467 is currently incomplete given that the course is still ongoing.

out the lectures. I also designed the midterm exam questions to carefully balance the level of difficulty and the amount of material covered. The midterm exam was described as being "*appropriate; challenging, yet fair*".

In summary, I consider myself well-suited for teaching computer systems courses, particularly, operating systems, distributed systems, and compilers. With the breadth of my research in computer systems, I am also interested in developing advanced courses that address recent topics such as big data analytics and distributed system reliability.

Appendix I: Teaching Experiences at University of Toronto

- Course instructor, CSC467 Compilers and Interpreters, Fall 2018
  Delivered three lectures weekly (42 students); designed the course website and the lecture slides from the ground up.

- Teaching Assistant, ECE344 Operating Systems, Winter 2018

- Teaching Assistant, ECE454 Computer Systems Programming, Fall 2017

- Teaching Assistant, ECE344 Operating Systems, Winter 2017

- Teaching Assistant, ECE454 Computer Systems Programming, Fall 2016

- Teaching Assistant, ECE344 Operating Systems, Winter 2016

- Teaching Assistant, ECE454 Computer Systems Programming, Fall 2015

Appendix II: Selected Student Feedback from CSC467

- *I think the course is really well run and I like the organization of the schedule.*

- *The instructor shows careful creation and preparation of lecture materials and presents the slides in a logical, timely yet easy-to-understand manner. Lectures are interesting.*

- *The instructor seems very knowledgeable of the relevant topics, and able to breeze through complex topics and point out some considerations to be careful about in the material.*

- *The scope of the midterm exam was appropriate; challenging, yet fair.*

- *Maybe you can teach CSC488 next semester, I think the CS department is still looking for instructor for that course? Not sure, but maybe!*