Teaching Statement

Joseph Izraelevitz
University of Rochester
Rochester, NY, USA
jhi1@cs.rochester.edu

1 Course Offerings

My background and research has focused on library-level systems software. Consequently, at the undergraduate/introductory level, I can teach courses in parallel and distributed programming, systems software, computer organization, data structures, operating systems, introductory programming, database management systems, and compilers. At the graduate/seminar level, I can teach courses in parallel and distributed programming, systems software, and operating systems. I would be interested in developing a new seminar class on advanced topics in shared memory programming, including topics in data structure design and performance evaluation. I would also be comfortable advising capstone team projects/undergraduate theses.

For teaching recommendations, please see those written by Profs. Michael Scott and Chen Ding.

2 Teaching and Mentoring Philosophy

My teaching and mentoring philosophy is rooted in three principles: enabling learning by streamlining tasks, structured explanation to allow students to follow the thread of instruction, and constant feedback on progress and expectations.

My first principle focuses on enabling the student to learn by streamlining tasks. Too often, students are distracted from the subject they are supposed to learn (e.g., the implementation of a specific graph algorithm) by secondary work that is necessary to complete but of little educational value (e.g., the implementation of the parser to read the graph data). When unchecked, these side issues cause countless problems: students are unable to complete the assignment because the secondary issue took too much time or was too difficult to accomplish, and consequently the students fail to learn the targeted topic.

While the teaching assistant for our undergraduate research seminar, during a unit on concurrent data structures, I provided a test harness for students which allowed for easy modification of test parameters and data structures, while also ensuring that the tests were correctly run. The harness allowed students to rapidly test and verify multiple data structure types, while avoiding exposing them to the many problems of correctly running multicore experiments (thread pinning, low impact timing, soft page faults on initialization, etc.) or their analysis (graphing, statistics management, etc.). As a consequence, students were able to correctly program interesting algorithms and analyze their results. This harness has since been used in five separate manuscripts, and has proved especially useful for enabling my mentored students to manage their research.

My second principle is structured explanation. Whether lecturing to a classroom or talking one-on-one with a student, I strive to structure my explanations into clearly defined sections that avoid forward references. Often, even while talking with a student individually, I will write out the outline of the topics we plan to discuss in order to frame the conversation. When appropriate, each topic we discuss is concluded with questions to verify that the students have understood the presented material before moving on to the next topic (e.g., by requesting the student rephrase a particular concept). While our university does not have formal TA evaluations, I have received emails of thanks for my teaching style: “I have been constantly finding this course to be very difficult . . . Thanks to you, I have been able to pick up anything that I do pick up in the course. I can see the improvement myself so I guess I am trying to say thanks!!”

My final principle is constant feedback. Students that I have supported or mentored are given expectations regarding milestones, and I strongly believe that structure is important for achieving positive academic
outcomes. This habit is a holdover from my experience as an officer in the U.S. Army, where periodic individual evaluations are the norm. I have found that, in general, good performance often follows from well-understood requirements, and poor performance is most commonly simply ignorance of expectations and status. All of my mentored students can expect weekly meetings with me where we discuss their current problems, brainstorm solutions, and make a plan for the coming week. Once a semester, this meeting also discusses their long-term and career goals in order to make sure they are managing their time wisely.

3 Teaching Experience

My teaching experience encompasses both in- and out-of-classroom learning. Inside the classroom, I have been the teaching assistant for three semesters: once for our grad-level Computational Complexity course (FL2013) and twice for our Undergraduate Research Seminar (SP/FL2014). As the TA for Computational Complexity, I was responsible for all grading for both tests and homework. While the primary instructor was out, I also gave a series of three full lectures on the class material, covering the Chomsky Hierarchy, with its associated grammars and machines, and Rice’s Theorem. As the TA for the Undergraduate Research Seminar, I was again responsible for most grading and for coordinating class projects. As this was a project-based class, I advised groups of students on their research projects in order to keep them on track throughout the semester. I also gave two lectures on my research to these classes.

I am comfortable lecturing. I have given talks at DISC’17, PACT’17, TRANSACT’17, DISC’16, SPAA’16, ASPLOS’16, TRANSACT’16, and SPAA’14. During the summer of 2016, while an intern at HPE labs, I gave an invited talk on my research at VMWare Research and a tutorial on the C++ memory model to my coworkers at HPE. I regularly lecture on my research to the wider systems community at the University of Rochester during our weekly seminar series.

I am also comfortable managing people. While in the U.S. Army, at Fort Carson, CO, I was responsible for the training and performance of my platoon of 15 soldiers on various technical tasks. As an officer, I was unlikely to directly instruct our soldiers unless it touched on an area in which I had particular expertise (this task is always the responsibility of platoon non-commissioned officers), but I was responsible for managing the training and adjusting the schedule based on feedback from NCOs. I believe my experience managing both training programs and people translates well into the classroom.

I have been the primary research mentor for three students, one undergraduate and two graduate. At the University of Rochester, several seminar courses give undergraduate and junior graduate students the opportunity to become involved in research with a senior graduate student, generally with minimal faculty supervision. This mentoring relationship often turns into a full-fledged research collaboration and continues beyond the end of the class. I have mentored three students through this arrangement.

Matthew Graichen began working with me as an undergraduate junior in the fall of 2014 on a project examining nonblocking double-ended queues. Our weekly hour-long meetings continued on into the fall of 2015. During this work, Matt developed a new nonblocking algorithm for unbounded double ended queues, designed a novel garbage collection scheme for it, and compared its performance to that of prior algorithms. We wrote up the paper together, and the final product was submitted and accepted at ICPP’16.

Hensen Wen began working with me in the first year of his Ph.D. program when I mentored him on his class project for our Parallel and Distributed Systems course during the spring of 2017. In this work, Hensen was implementing prior art in concurrent trees but became interested in new memory management schemes for them. The final project attracted several additional co-authors and will appear at PPoPP’18.

Wentao Cai also began working with me in the spring of 2017 during the first year of his master’s program. His work, again a course project for the Parallel and Distributed Systems class, focused on examining existing designs for concurrent trees. Together, we are incorporating these algorithms into a new framework for building lock-free concurrent trees, and expect to submit to SPAA’18. Due to his positive experience working with me and the rest of our lab, Wentao decided to switch over to the Ph.D. program at the University of Rochester.

Based on these experiences, I believe I have demonstrated an ability to bring new students up to speed in a research environment and to ensure that their work is engaging, meaningful, and productive. I am looking forward to a new job where I can continue working with bright students both in- and out-of the classroom, teaching them the art of research and parallel programming.