Moustafa AbdelBaky Teaching Statement

The ability to teach and mentor students is one of the most rewarding aspects of an academic position. One of the things that excites me the most about the field of computer science is that it is constantly evolving in response to new technologies and research – it is truly a *lifelong learning* environment and I am eager to share it with my future students. My goal as an educator is to help students build the skills needed to develop a strong theoretical computer science background, apply their knowledge in new settings, teach them how to learn new concepts on their own, tackle real-world problems, think critically about new ideas, and effectively communicate their work to others.

Experience and Mentorship: As a graduate student, I was not afforded the opportunity to be a formal teaching assistant or lecturer. However, teaching others is extremely important to me and so I sought out external educational experiences that would allow me to do so. Specifically, I have been an instructor for a summer school online-based course on federated cloud computing, given two full-day conference tutorials on cloud computing, led group tutoring sessions, and participated in one-on-one mentoring.

For the full-day conference tutorials, I spent several weeks preparing slides, documentation, and in-depth code examples that showcased all of the features of the software. I delivered the tutorials as a mixture between lecture-style presentations and hands-on exercises. After the first tutorial, I received positive feedback from participants on the conference evaluations but their suggestions varied from "make the course more technical" to "spend more time on the introductory examples." Based on this feedback, I came up with the idea to expand each hands-on exercise to include individual components that target multiple skill levels (beginner, moderate, advanced). In the lecture portions, I focused on the main takeaways of the exercises and necessary details like the API calls rather than reading a slide with lines of code on it. This enabled advanced learners to continue at their own pace when more attention was needed from beginners. Ultimately, the feedback I received from the second tutorial was excellent and more participants stayed for the entire day rather than coming in for a short while and then leaving. I am eager to apply this method to semester-long courses and think it is a good fit for large class sizes, due to the greater variation in skill levels.

During my time as a graduate student and postdoc, I also had the privilege of mentoring undergraduate and graduate students in both educational and research settings and am eager to continue doing so as a future professor. In mentorship sessions, one of the most common issues I have encountered is that there is a disconnect between the structure that completing coursework toward a specific degree provides and the open-ended nature of research. This experience made me realize that more needs to be done to promote research at an undergraduate level so that students have an awareness of what research is and how they can get involved with research in the department.

One way that I have encouraged student involvement in my own department is by inviting graduate students and other postdocs to collaborate on a survey paper I am writing about cloud, fog, and edge computing. Rather than assigning papers to students and asking them to independently write literature reviews, I decided to organize a weekly journal club style meeting, where different students take turns presenting 1-2 papers each week and then we discuss the paper's research critically as a group. This format not only introduces students to cutting-edge research and effective ways of communicating technical ideas but also encourages collaboration between them on their own research ideas. As a future faculty member, I look forward to offering these types of opportunities within my department while also identifying other ways to improve the student experience.

Teaching Philosophy: Reflecting on my own experience as a student and inspired by multiple successful faculty who promoted active learning, my teaching philosophy is to 1) alternate class time between content-focused lectures (on theory and abstractions) and hands-on in-class assignments that draw on real-world examples, 2) develop well-structured research-oriented projects for undergraduates, and 3) utilize technology in the classroom such as collaborative notetaking and Jupyter notebooks.

When delivering lectures, I am mindful of the fact that students learn in a variety of ways and so I try to approach the conceptual content using multiple teaching modalities. For example, I will utilize visual illustrations or animations of an algorithm in addition to analysis of the pseudo-code. I will also incorporate kinesthetic learning when possible, e.g., demonstrating distributed mutual exclusion by using a small group of students as "processes" and having them physically act out a message passing "scene" for the class. In addition to varying the teaching methods used, I also aim to connect the theoretical content of the course to tangible real-world applications like scientific research or pop culture (e.g., clustering algorithms can be of great importance in image detection applications for cancer research but are also used by the Xbox Kinect to estimate the joint positions of a player in the game "Just Dance").

In large lectures where it can be difficult to gauge students' overall grasp of the content, I plan to utilize collaborative notetaking during lecture sessions, e.g., a common Google Doc for the lecture that all students have access to and can update. This strategy has been used at Berkeley where enrollment in introductory computer science courses exceeds 2000 students per course. Collaborative notetaking helps students stay engaged in class and promotes an environment where peers can directly assist other peers in real-time. Additionally, after class is over, this document will be useful to me in gauging what the overall student takeaways were from a lecture versus the overall goals of the content that the lecture was intended to cover – and what points I should go over in the next lecture.

For in-class assignments, I will use pre-annotated python notebooks in Jupyterlab to demonstrate the core concepts covered in the lecture programmatically. I will also use a second notebook containing in-class exercises that will be submitted for cursory evaluation of whether the student achieved the correct answer or not. On these exercises, I will vary the difficulty of questions over a range of beginner, moderate, and advanced to keep students of all levels interested and then use that data to adjust the content covered by the pre-annotated notebooks. Further, I plan on combining these in-class exercises with weekly (or biweekly) project-style homework assignments, which provides the opportunity to connect multiple class concepts to their broader significance in science or technology domains (e.g., after a lecture on binary search trees, students implement one and evaluate the time it takes to find specific sequences in large-scale DNA datasets).

Courses I Can Teach: At the undergraduate level, I am interested in teaching courses on introduction to programming languages, data structures and algorithms, principals of programming languages, networks, and operating systems. At the advanced undergraduate and graduate level, my background makes me well-suited for teaching parallel and distributed computing and software engineering. I am also interested in developing seminar-based courses on 1) blockchains vs. traditional distributed systems and 2) cloud-edge computing and their role in the Internet of Things. In these courses, students will combine critical analysis and discussion on cutting-edge research papers with a focused semester-long project. I would also love to teach a course on technology entrepreneurship, a joint class with the business school, where students from both departments work together on a semester-long project to create a product and learn all the necessary steps in building a startup. Lastly, I am interested in creating a course on the ethics of artificial intelligence and distributed systems and their role in democracy. This course provides a great opportunity to teach students about the role of computer systems in society and will also inform my research moving forward.