Teaching Statement

Amir Kamil

January 31, 2015

I started my teaching career rather reluctantly. Growing up, I always wanted to be an academic, but I envisioned myself as a researcher rather than a teacher. I was terrified of public speaking and had never given a talk of more than a few minutes. While I participated in my high school's tutoring program and enjoyed interacting with individual and small groups of students, I didn't think I was capable of holding the attention of a full class for an hour or more. Unfortunately, I realized in college that professors have to teach, so I decided to see what I was in for if I chose a career in academia. So with great reluctance, I signed up to be an Undergraduate Student Instructor (UGSI) in the summer of 2002 for CS61B, the data structures course at UC Berkeley, after just two years as an undergraduate.

I walked into my first discussion section as a nervous wreck, wondering what I had gotten myself into. Standing in front of a classroom of 25 students, I looked up at the clock, thinking there was no way I could keep their attention for the next fifty minutes. They had just had their first lecture, much of which consisted of administrative details, so I didn't even have a lot of material to work with. I began hesitantly, introducing myself and covering a few more administrative issues. I then spent a few minutes giving the students advice; I had taken the class the summer before, and I described what to expect from the course and how I coped with the fast pace in the compressed summer timeframe. I then dove into the basics of object-oriented programming and Java, going over simple examples and answering questions. By the time I lookup at the clock again, the fifty minutes were up; to my surprise, I had not only survived my first class, but I had thoroughly enjoined it. I couldn't wait for my next session, my nervousness completely replaced with excitement and anticipation. In fifty minutes, I had gone from dreading teaching to loving it.

Though I was also taking two classes that summer, I spent most of my time teaching, whether helping students on their assignments at odd hours down in the lab or working with the instructor, Barath Raghavan, to develop course projects, homeworks, and exams. I followed up by teaching the class again in the fall for Professor Paul Hilfinger, and in the spring of 2003, I was the head teaching assistant for Professor Katherine Yelick in CS61B, playing a key role in creating assignments and managing the teaching and grading staff. In the fall of 2003, I was the sole teaching assistant for Professor David Wagner in CS70, the discrete mathematics and probability course at Berkeley, in which I managed the grading staff, wrote homework solutions, and contributed exam questions in addition to running discussion sections. In my last semester as an undergraduate in spring 2004, I was a UGSI for CS162, the upper-level operating systems class. I spent a lot of time maintaining the course infrastructure, impressing the instructor, Professor Anthony Joseph, to such an extent that he hired me in the summer to overhaul and modernize the course software and projects. Thus, I taught five times and spent an additional summer on course development before I

started graduate school.

While it was now clear to me that teaching was my true passion rather than research, on the advice of Professor Yelick, who had become my graduate advisor, I focused on research and making progress toward my dissertation in the first few years of graduate school at Berkeley. It wasn't until 2009 that I taught again as a Graduate Student Instructor (GSI) for CS70 for Professors Satish Rao and David Tse. That experience rekindled my love of teaching, and I insisted on continuing to teach, even though I knew it would delay my graduation. I again taught CS70 in the fall of 2010, this time as the head teaching assistant for Professors David Tse and David Wagner. For my contributions that semester, I received an Outstanding GSI award from UC Berkeley as well as an honorable mention for the EECS department Outstanding GSI award.

Though I enjoyed my many semesters as a UGSI and GSI, I wanted to move on to the next challenge as the primary instructor for a course. In the summer of 2011, I had the opportunity to teach CS70 as the instructor of record for a class of about sixty students, responsible for everything from putting together a syllabus, hiring a teaching staff, lecturing, creating assignments, writing exams, and assigning grades. Despite the compressed schedule and the fact that it was my first experience as the primary instructor, the class went very well.

After taking another year to focus on completing my dissertation, I was hired as a Lecturer at Berkeley to teach CS61A, the introductory Computer Science course. This proved to be a new challenge, with 650 students across two lecture sections, eleven teaching assistants, ten graders, and dozens of volunteer lab assistants. I had never experienced a class of this size, and I had no experience with CS61A aside from taking it twelve years earlier. In the meantime, the course had been overhauled with a new language that I did not know (Python), a new text, and new projects. In spite of these challenges, the course went well, and I even had the opportunity to experiment with new venues for interacting with students (the study hall and help sessions mentioned below) as well as create a new introduction to parallel computing.

Though I was offered the opportunity to continue as a Lecturer at UC Berkeley, I chose instead to move to Lawrence Berkeley National Laboratory (LBNL) as a Postdoctoral Fellow so that I could gain experience working at an institution other than UC Berkeley. However, I accepted the LBNL position with the understanding with both my supervisor at LBNL and the UC Berkeley CS department that I could spend some of my time teaching at UC Berkeley. Initially, I planned to co-teach a graduate-level parallel languages course with Professor Yelick. To my disappointment, we were unable to get approval from the Human Resources department at LBNL after many months of trying, and I could only be involved in a peripheral and unofficial capacity. While I appreciate the opportunity I've had to do research and work with new people at LBNL, I am looking once again to change my focus to my true passion of teaching.

I am still at the beginning of my teaching career, and though I've had the opportunity to work with and learn from many great teachers at UC Berkeley, I am constantly striving to improve my teaching skills and further develop as an educator. As my course surveys demonstrate, for each class I've been involved with, my ratings have improved in each subsequent semester as I've gained a deeper understanding of the material and a better idea of how to present concepts to students. This year, I am attending the ACM Computer Science Education (SIGCSE) conference for the third time, spending my own money to further develop my skills and learn from the top Computer Science educators. I look forward to continuing to improve as a teacher, and I hope I can inspire my own students to pursue careers in Computer Science education.

Teaching Philosophy

As a Computer Science educator, I try to bring my love and enthusiasm for Computer Science into the classroom. I've found that enthusiasm is contagious, that students pay more attention when they see that the instructor is genuinely excited about the material. I also try to pique students' interests by using examples that are tied to familiar real-world examples and popular culture. For example, I've demonstrated recursion by referencing the multiple dream levels in the motion picture *Inception* (with Limbo as the base case), an episode of the television show *Futurama* in which a character recursively creates smaller copies of himself, and an elaborate (fictional) story where I try to impress a girl by lifting a box of books, but since I'm too weak, I have to keep removing books until it is light enough to lift. Of course, I've found humor, particularly of the self-deprecating variety, to be a valuable tool in keeping students' attention.

My teaching style can be best described by the following quote from a student:

> Kamil is one of the best instructors I have ever had (whether I pass the class or not :)). He has a "tutoring" style of teaching that I like, rather than "instructing." Very accessible; I ask a question during office hours and he lowers the explanation incrementally until I understand it.

When teaching a larger group of students, I like to give them insight into my own thought processes and how I approach a problem. Often, when interacting individually with a student, I prefer to flip this procedure and ask the student to use his or her thought process to guide me through a problem. I've found these techniques help in understanding the mental processes that lead students astray, so that I can in turn help them develop better mental models and approaches that lead to a correct solution.

While I'm perfectly happy lecturing to any number of students, I find interactions with individual or small groups of students to be the most rewarding. As such, I try to make myself available to students as often as possible in many different settings. In addition to standard office hours, I encourage students to find me after lecture, on many occasions staying for half an hour or more to answer individual questions. I experimented with a "study hall" environment where students could come to work on assignments, with teaching staff, including myself, available to provide individual help. My staff and I also held help sessions just before each project deadline where we worked with students to debug their code and help them complete their projects on time.

In my experience, students often learn the most from their peers, whether their classmates or those who have recently taken the same class. So in addition to encouraging a collaborative environment in office hours, study hall, and help sessions, I try to create as many opportunities as possible for undergraduates to contribute to the teaching experience. This includes hiring undergraduate teaching assistants and graders when possible, but also accepting volunteers to help other students in office hours, labs, and discussion sections in exchange for course credits. Often, these volunteers later move on to paid positions as graders and teaching assistants, with their experience interacting with students and staff helping to develop their teaching skills. Similarly, I allow teaching assistants to give one lecture during the semester and involve them in all aspects of course management, including writing exams, developing projects, making scheduling decisions, and assigning final grades. Such a teaching environment benefits everyone, with the instructor having more time to give attention to individual students, former students remaining invested in

the course and developing their teaching skills, and current students having many opportunities and venues for seeking help and learning the material from a variety of viewpoints.

Diversity in Computer Science

I believe that all students benefit from a diverse learning environment, and my own experience as an undergraduate bears that out. When I signed up for my first Computer Science class at UC Berkeley, I had little confidence in my ability to succeed, as I had no formal education in Computer Science and did not identify with what I thought of as the typical Computer Scientist. Thankfully, my understanding of a typical Computer Scientist was upended as soon as I attended my first lecture and a Latino lecturer (the great Dan Garcia) took the stage, and then went to a lab section taught by a woman. Though I am neither Latino nor female, experiencing such a diverse teaching staff helped me overcome my insecurities about whether or not I belonged in Computer Science, and I believe that many students would similarly benefit from having diverse role models.

I have also observed that the lack of a diverse environment can lead to marginalization, particularly for female and minority students. In my experience as a teaching assistant and instructor, I have noticed that women in particular are much more comfortable participating in class and asking questions when there are more women present than when most of the students are male. I have also seen women marginalized in group projects, where a team dominated by men refuses to let the female members contribute in any significant way to the project. I believe that greater female and minority representation will make for a more comfortable and welcoming environment and discourage any student from being marginalized.

As a teacher, I consider it to be my responsibility to foster a more welcoming environment for all students. The following are some strategies I've used to do so:

- When teaching a class, I make sure that everyone is involved in the discussion and do my best to prevent anyone from dominating it. In office hours, I make sure to divide my time and attention equally among all students, and if a student does not proactively ask a question, I make sure to go to the student individually and ask if he or she needs any help or has any questions.

- When lecturing or writing an assignment, I like to use examples from popular culture that students can relate to. Since many students' interests differ from my own, I try to use a broad variety of cultural references, going as far as watching popular movies and television shows that I have little interest in to diversify the references I make in class.

- For project courses, I've found that asking students to confidentially evaluate their team members and describe how the work was divided among the team helps identify problems earlier in the course. This allows the teaching staff to work with a team to resolve these issues before they damage a team beyond repair.

- When putting together my teaching staff, I take diversity into account to the extent allowed by the law and departmental policy. I also encourage female and minority students to pursue undergraduate teaching opportunities to increase the diversity of the teaching staff in future semesters.

While these strategies have proven beneficial, I believe there is much more that can be done to address the needs of underrepresented students, and attracting more female and minority students to Computer Science is one of my major goals as a Computer Science educator.

Teaching Interests

As can be seen from my experience, I am comfortable teaching a wide variety of courses ranging from core introductory subjects to advanced programming systems. More specifically, my main interests lie in the following areas:

- Core introductory courses including introduction to programming, data structures, and algorithms.
- Theoretical courses such as discrete mathematics, probability, and computational theory.
- Programming languages, compilers, synthesis, and program analysis.
- Parallel and scientific computing.

I am particularly interested in developing an introductory parallel computing course for undergraduates. Parallel computing has been a major topic of my research, but I feel that the Computer Science community does not do an adequate job teaching it, despite its increasing importance in the real world.

Teaching Evaluations

The following are selected quotes from anonymous course surveys for CS61A at the end of spring 2013:

> His enthusiasm and attempts to be funny keep the lectures interesting.
>
> Professor Kamil was very accessible, I remember when he was in Soda Hall on the night the Hog project was due, just to ensure anyone who needed help would get it.
>
> One of the most hardworking profs I've seen. Extremely prompt with Piazza questions.
>
> Fun. I really enjoy how he includes fun examples in lecture, such as references to Inception, Starcraft, etc.
>
> He explains topics VERY clearly and is very dedicated to the class and students. Enthusiastically encourages questions and is very knowledgable about the subject matter. Always provides answers that are easy to follow.
>
> Prof. Kamil is an awesome professor! He's interesting, helpful, clear, down-to-earth, and a great teacher. Not only does he make the material straightforward, but also fun. Very lucky to have had him this semester!

> Incredibly enthusiastic and a highly effective presenter. Breaks things down into manageable chunks and continually connects the low and high level information. Explanations are consistently lucid, and enjoyable.
>
> I know Amir has not spent much time on his own research because he has devoted himself to this course and its students w/ perpetual enthusiasm and accessibility. He is an incredibly good course leader/lecturer and I am grateful to him and 61A for such a wonderful experience.
>
> Loved the instructor! He is a very nice guy and explains concepts clearly! Easily accessible outside class hours. Knows his stuff in and out!
>
> One of the best professors. Always available, easy to talk to, very encouraging. Told me not to drop the class. Glad I didn't, learned so much.

The following page is a summary of my ratings each semester[1] I taught at Berkeley. Full ratings are available at `https://hkn.eecs.berkeley.edu/coursesurveys/instructor/6910`.

[1]Note that the ratings for spring 2013 on the next page and the HKN website as of this writing are based on incomplete survey data. The effectiveness score of 5.6 cited in my CV is computed from the full set of data on file with the CS department.

Amir Kamil

Title	Teaching Assistant
Office	566 Soda
Phone Number	(510) 642-8493
Email	kamil@eecs
Home Page	http://www.eecs.berkeley.edu/~kamil

Classes Taught

Sections	Teaching Effectiveness	How worthwhile was this course?	Other Instructors
CS61A Spring 2013	5.5 / 7	6.2 / 7	
CS70 Summer 2011	5.9 / 7	5.7 / 7	
Totals	**Teaching Effectiveness**	**How worthwhile was this course?**	
CS61A (1)	5.5 / 7	6.2 / 7	
CS70 (1)	5.9 / 7	5.7 / 7	
Undergraduate Courses (2)	5.7 / 7	5.9 / 7	
Graduate Courses (0)			

Classes TA'd

Sections	Teaching Effectiveness	Instructors
CS70 Fall 2010	4.6 / 5	David Wagner, David Tse
CS70 Fall 2009	4.2 / 5	Satish Rao, David Tse
CS162 Spring 2004	3.8 / 5	Anthony D. Joseph
CS70 Fall 2003	4.0 / 5	David Wagner
CS61B Spring 2003	4.5 / 5	Katherine A. Yelick
CS61B Fall 2002	4.5 / 5	Paul N. Hilfinger
CS61B Summer 2002	4.0 / 5	Barath Raghavan
Totals	**Teaching Effectiveness**	
CS61B (3)	4.3 / 5	
CS70 (3)	4.3 / 5	
CS162 (1)	3.8 / 5	
Undergraduate Courses (7)	4.2 / 5	
Graduate Courses (0)		