Reflective Statement

John DeNero

The demand for introductory computer science and data science education at UC Berkeley has grown tremendously in the last decade. In Fall 2017, I co-instructed two courses: *Structure and Interpretation of Computer Programs* (CS 61A) and *Foundations of Data Science* (Data 8). When I started teaching in 2011, the former had 534 students and the latter didn't exist. In Fall 2017 they had a combined enrollment of over 2,700 students and a combined staff of 166 student instructors and tutors.

I strongly believe that our students should have the opportunity to take these courses when they choose, but scaling our course offerings to keep up with this surge in demand has required substantial changes. I also believe that the way we teach computing should improve, even as these courses grow. I'm proud to say that so far, I have never had to turn away more than a handful of students from any of my course offerings, and student feedback is continuing to improve. The last time I taught *Foundations of Data Science* alone (Spring 2017), I asked students, "How happy are you about your decision to take Data 8?" 80% (547/680) responded 4 or 5 out of 5. When I asked the same question about CS 61A the last time I taught it alone (Fall 2016), a record high of 84% (1,206/1,432) responded 4 or 5 out of 5.

But my goal in teaching is not to make students happy. What I really want is to help students learn how to solve problems that they thought they couldn't solve. When that happens, the effect is transformative. They build confidence and enthusiasm in addition to understanding. I'll share three examples of how changes in teaching practice have simultaneously helped scale up enrollments and helped foster more of these critical moments of surprising achievement.

As I often tell my students, the purpose of taking a course is to learn something new, not to demonstrate what you already know. For this reason, I have designed large-scale projects for both courses that challenge students to combine multiple concepts and build something substantial—some result that stretches students' capabilities beyond their expectations. There's a risk to this approach. Completing an ambitious course project can inspire and motivate students, but struggling for hours without progress offers little benefit and a lot of frustration. Therefore, much of my effort as an instructor has been to scaffold these large projects with an automated feedback system[1], so that students know whether they are making progress. One of my favorite features of this feedback system is that students interactively explore the automated checks that will be applied to their solution before they try to solve each problem. We found that this intervention drastically reduced the number of clarification questions posed to the course staff. This one subtle nudge toward a more effective process of problem solving saves hundreds of hours of instructor and student time each semester. It has also nearly eliminated the frustration some of our students used to face after spending hours trying to solve a problem, only to realize that they were supposed to be solving a different problem all along. Changes like this one allow more students to successfully complete large course projects and gain the confidence to start projects of their own.

The activities of my course staff have also evolved dramatically, often due to the leadership of dedicated student instructors. For example, the course tutors who used to focus exclusively on grading and drop-in office hours now also lead weekly small-group tutoring sessions for up to four students at a time. They formed a new student-led organization called CS Mentors to schedule and coordinate these sessions. In Fall 2017, over 800 students regularly attended these weekly tutoring sessions across my two

[1] http://okpy.org

courses. Attending these tutoring sections had a significant positive association with exam scores. I find the scale and impact of this tutoring effort to be inspiring. I don't take credit for it, nor for the brilliance and dedication of the members of my course staff, but I am proud to have brought a critical mass of these talented students together and encouraged them to pursue ambitious goals.

A final important change has been video recording and web broadcasting, which was essential to scaling beyond the capacity of campus lecture halls. Now that all of the lecture content for CS 61A has been recorded, I have the opportunity to focus on support material. One of my favorite uses of video is to broadcast explanations of assignment questions to all students, which can save some of them a trip to office hours. First, I hold live office hours to learn what questions are shared among multiple students. Afterward, I record a short video answering those questions and post it for the rest of the class. The most recent such explanation was viewed 905 times, even though fewer than 10 students came to my live office hours that day. This approach lets me reach a large number of students, but still work directly with individual students through that critical process of wrestling with a daunting problem until that sublime moment when they discover that they can solve it after all.

I believe that only at Cal could we have maintained this scale of enrollment growth in computing courses while consistently improving our students' learning experience. Faculty from across campus, a multitude of student instructors, and an incredibly supportive staff and administration have all collaborated to make Berkeley a world leader in undergraduate peer instruction, data science education, and computing education in general. It has been my great privilege to participate in this achievement and to work with so many of the brightest people in the world.