

Teaching/Advising Statement

David J. Malan
malan@harvard.edu

My views on teaching have not only crystallized but also evolved over time. When I first began, I'm not sure I had an overarching philosophy. Instincts and my own past experiences as a student alone guided my pedagogical decisions. But I was sensitized early on, by way of teaching at DCE especially, to students “less comfortable,” for whom some subject did not come easily. And over time, patterns among my pedagogical decisions emerged, perhaps informed by my own doctoral research, which focused on detection of patterns in large (albeit very different!) datasets. Ever since, I daresay three principles have guided my teaching: accessibility, rigor, and community. These are particularly manifest in CS50, given its size, both on campus and off, but they also characterize my teaching at DCE, HBS, HLS, and beyond.

In years prior, I might have described my views on teaching as very “libertarian,” too, with my courses designed to meet students halfway, with students free to engage with the course in a manner consistent with their own learning style. In CS50, for instance, for years, students were encouraged but not required to attend lectures in person; they could alternatively watch online afterward instead. The course's several policies (e.g., on academic honesty) have long been in the course's syllabus, and announced as such, too, but it was on students to read them.

Over time, though, perhaps with more experience (or just age!), I have come to appreciate more, particularly at scale, the pragmatic value of helping students help themselves, as via some additional structure. For instance, we now expect students in CS50 to attend lectures in person (and take attendance upon entry to emphasize such), unless simultaneously enrolled in another class (or academic or athletic activity).¹ We otherwise found that too many students would intend to, but not actually, watch lecture before section, thereby arriving at the latter underprepared. In turn, we would see more questions in office hours, a problem that's only magnified at scale. We also now have a start-of-term “quiz” on the course's policy on academic honesty, to increase the probability that students not only read but reflect on it as well.

When it comes to advising outside of the classroom or, perhaps even more often, right after class, I find that I draw inspiration from my own educational experiences and also regrets. I consistently advise students to take CS50 SAT/UNS (except when their concentrations require otherwise) so that, with higher probability, they focus on each week's lessons more than they do each prior week's scores. I myself, I tell them, took CS50 Pass/Fail, only

¹We have not found that simultaneously enrolled students underperform vis-à-vis students who attend lectures in person.

switching to a letter grade at the last minute for concentration's sake. I encourage students to make room in their schedule, early on, for exploration of courses outside their comfort zone that might lead them toward interests they didn't realize they would have. I myself, I tell them, only found my way to computer science because I shopped CS50. And some of my favorite classes (in Latin, dramatic arts, archaeology, and physics), I tell them, I didn't take until senior year or even graduate school, at which point it was too late to pursue follow-on courses within my schedule as well.

My own experience is surely not necessarily theirs, but, all these years later, these life lessons remain forefront of mind. And so the advice about which I'm most passionate tends to be that which I wish I myself had received.

Accessibility

It's important that introductory courses be accessible, which is not to say easy but, rather, within academic reach of all students, irrespective of prior background or lack thereof. Quite often do introductory courses serve multiple audiences: non-majors for whom the course will be terminal, their only exposure to a field, and majors for whom the course is a gateway to higher-level studies. Barriers to entry serve neither demographic. If the on-ramp to a course is too steep, with the course assuming too much prior knowledge or accelerating too quickly without an adequate support structure for those less comfortable, non-majors might steer clear altogether and even prospective majors might veer off their initial trajectory. In both cases is our field poorer for it, with fewer of its ideas in the wild and fewer scholars trained to apply its lessons to their field or ours.

While it might be tempting to offer each demographic its own on-ramp, as by offering one course for non-majors and another for majors, such an early fork in the road forces students to choose a priori their destination, even before many have any experience based on which to make an informed decision. A premature fork risks creation of a class of students who assume they do not belong in the other.

It's for precisely this reason that we continue to keep students less comfortable and more comfortable alike in CS50 and provide for the former especially a robust support structure so that all students, irrespective of prior background or lack thereof, can succeed, provided they avail themselves of those resources. And it's for this same reason that we encourage so many students to take the course SAT/UNS so as to explore an unfamiliar field without fear of "failure."

Rigor

Accessibility and rigor, though, are not mutually exclusive. In CS50, for instance, we expect no less of those less comfortable than we do of those more comfortable. Indeed, among General Education courses in the sciences alone, wherein workloads average just 4–5 hours per week, CS50 expects upwards of 12 hours per week of most students. The course does not shy away, either, from material that most introductory courses defer to higher-level courses. By CS50's sixth week, for instance, students are implementing their own hash tables in C, and two thirds of those students have never taken a CS course before.

But that rigor is made possible via the course’s support structure, which includes weekly sections, daily office hours, and nearly 24/7 support online, all thanks to the course’s team of teaching fellows and course assistants (with wide-ranging sleep schedules) and the course’s own software for teaching and learning. Among those sections, meanwhile, do we have different tracks for those less comfortable, those more comfortable, and those somewhere in between, tailored to each demographic’s background. And half of the course’s problem sets offer less-comfortable and more-comfortable variants of problems. While both assess students’ mastery of similar material, one offers a lower floor and the other a higher ceiling. Students are free to choose one or the other (or both) for equivalent credit but are encouraged to reach higher as the semester progresses.

By scaffolding a course in this way, with training wheels on at term’s start and off after distances that vary among students, students’ progress can be measured vis-à-vis students’ own starting points, with the course’s rigor thus relative. Per CS50’s own syllabus, “what ultimately matters in this course is not so much where you end up relative to your classmates but where you end up relative to yourself when you began.” On display at each CS50 Fair, meanwhile, is the result of taking those training wheels off. Via final projects do students ultimately learn how to teach themselves something new, without any specification from us. Indeed, at term’s end, a successful outcome is when students have learned something that we did not teach them.

Community

Perhaps most underappreciated in higher education is the opportunity to transform courses themselves into communities, each with its own culture and norms, a shared experience wherein students might collectively struggle but, with the right support structure, succeed. Indeed, a course needn’t be just a course. It can instead be its own support network wherein students know (and are reminded) that they have classmates, just as uncomfortable as they, on whom they can lean as they progress together toward the finish. That so many of CS50’s own TFs and CAs are not only undergraduates themselves but also alumni of the course fosters community too. And most every outward manifestation of CS50, including its t-shirts and stress balls, its lunches and office hours, its hackathon and fair, is also intended to instill a sense of community and, in turn, collective accomplishment. It’s for this reason, too, that CS50’s t-shirts say at term’s end, simply and proudly, “I took CS50.”