Statement on Teaching Philosophy

Hammad Ahmad

I believe that it is important for educators to make sure that all students are adequately supported, and I strive to create a learning environment that promotes academic belonging and reduces barriers for success. During my time teaching as instructor of record across two different institutions (University of Michigan and Washington and Lee University), and through several years professional development programs, I have honed my teaching style and philosophy to provide equitable and inclusive instruction to students with varying backgrounds and identities.

Teaching Philosophy

Instead of attempting to encapsulate my entire teaching philosophy into one document (lest I keep pontificating for dozens of pages), I will focus on summarizing my approach to teaching with the following overarching themes: cultivating a growth mindset, administering adaptive lectures, and keeping students engaged in the classroom.

Growth mindset. I firmly believe that people can get better at things over time, and most differences we see in student outcomes are not because of inherent properties but because of incoming preparation. As such I strive to give my students enough time and feedback to learn and grow into the course objectives, and I make a deliberate effort to give students feedback on an assignment well before the next assignment is due. This effort requires careful scheduling of my time to ensure students get timely and thorough feedback, but I find it helps tremendously in giving people a chance to improve at the course. I also put a strong emphasis on normalizing being wrong, since I find it to be helpful in cultivating a growth mindset among students, and that students (particularly non-traditional learners with less incoming preparation) are often encouraged when they see they are not alone in their struggles. I have also developed my teaching style to incorporate an informal tone during lectures and office hours, since I have found that students are less afraid to be wrong in a lower stakes, casual setting.

timely feedback to students

My experience teaching introductory and intermediate courses as instructor of record has unveiled to me the wide range of incoming preparation that students have prior to taking college courses,[1] and I design my courses with this preparation gap as a focal point in four ways: *reaching out* to struggling students, *being available* outside of class, *adapting* equitable and flexible course policies, and *incorporating* inclusive vernacular. First, I have found that students who are falling behind are often also those with less incoming preparation for Computer Science, and as a result of imposter syndrome or not feeling a sense of academic belonging, are more likely to drop the major. As such, I frequently reach out to students who may otherwise be left behind, including students who unexpectedly started missing assignment deadlines, were consistently getting grades lower than 2 standard deviations of the mean, or were not contributing during in-class activities during lectures. I have found that such students often have external, non-academic reasons behind their struggles, and instead of putting the burden on them to perform the uncomfortable task of approaching a professor, I make an effort to reduce the anxiety associated with asking for support by initiating a low-pressure email conversation. Second, I regularly make myself available for asynchronous questions outside of class and office hours, particularly before assignment deadlines and exams. Doing so requires that I iron out any kinks with the exam or assignment well in advance, but the opportunity cost is easily justified to me since I have found that most students often need last-minute help with the course materials before a major deadline (particularly students who are unable to get an early start on the assessment due to external factors). Additionally, I have observed that being available outside regular hours offers a benefit to students who have scheduling conflicts (academic or otherwise) and may not be able to get help at scheduled times. My former students have reported:

reaching out to students

availability outside class

[1] Indeed, this realization has led me to explicitly investigate the role of incoming preparation in achieving positive outcomes for Computer Science tasks.

– Professor Ahmad . . . was very easy to reach outside of class . . . .
– . . . he did an excellent job explaining everything and making sure we understood everything. He also made himself available and was very helpful during labs! Kudos!
– The professor was always available for questions and provided guidance to the students.
– Awesome course. Professor is really reachable and helpful.

equitable, flexible policies

Third, I put an emphasis on equity over equality for assessments and grading, and am a proponent of flexible course policies. As a concrete example, I chose to give extra time to a student to turn in a programming assignment and an online quiz since they did not own a laptop (and, *unbelievably*, the library ran out of borrowable laptops). While the student could have worked from a library computer to complete the assessments, I do not believe that a student should be at a disadvantage for not owning a computer. Similarly, I chose to offer makeup assessments for a student who started missing deadlines due to the constraints of picking up a second part-time job due to financial difficulties. I do not believe that the student, who had historically submitted assignments on time and had reached out to me explaining their situation, should be penalized for circumstances outside their control. Both cases are indicative of my efforts at designing and structuring my courses with internal scaffolding guidelines that encourage students to stay on track with the course (e.g., posting assignment deadlines before the start of the semester), yet offer students an equitable chance to demonstrate their mastery of the course materials. Finally, I have adopted certain vernacular in my teaching that I believe promotes an inclusive learning environment. For instance, I have made efforts to replace statements like "we saw this content before in week 3" with "some of us may have seen this before in week 3, but let's revisit this together for a moment", since the latter avoids students who missed class or do not remember the materials feeling left out.

inclusive vernacular

Adaptive lectures. The second major theme of my teaching philosophy revolves around administering adaptive lectures. On the one hand, I favor dynamic and adaptive lectures as a medium of instruction. During my time as an instructor over Zoom, I found that pre-prepared (static) lecture slides offer an accessibility benefit in that students can download them for review before and after a lecture, students who miss class or non-native speakers of English can use published slides to catch up, and so on. But on the other hand, my experience administering in-person lectures has made me appreciate the dynamic nature of digital whiteboards that allow a lecture to be tailored in real-time to student questions and needs. For the last two courses that I taught as a primary instructor, I was able to incorporate the "best of both worlds": I use a tablet to project pre-prepared lecture materials with a blank page every 3-5 slides, and annotate frequently on the slides during the lecture. The blank slide offers me space to answer impromptu student questions and tailor the lecture as needed (e.g., if student discussions necessitate more clarification or conversation on a certain topic). I have also found that the forced break every few slides helps me pace my lectures, and offers students frequent opportunities to ask clarification questions.[2] I publish both the original and the annotated versions of the slides after each lecture for students to reference the course material later.

dynamic instruction medium

formative assessments

In addition to steering individual lectures based on student discussions and questions, I put an emphasis on using formative assessments (e.g., low-stakes, announced weekly quizzes, asking students to summarize key concepts from the week's lectures, cf. [1]) to guide my course design and lecture preparation. The formative assessments not only offer a chance to students to review and keep up with the course materials, but also help me understand the gaps in the students' knowledge and allow me to address those gaps (e.g., by re-structuring lecture and office hours time) before we move on to other topics. I have also found that letting my students know that questions appearing frequently for such formative assessments may well reappear for the final exams encourages the students to keep up with the assessments. This is exemplified in the following student comment after the course's final exam:

> *I didn't like [weekly quizzes] at first since they increased the workload for the course, but I ended up realizing that they helped me study for the final all along . . . and not fall behind [in the course].*

course design

I am also a strong proponent of designing courses around student-centered, measurable learn-

[2] I recognize that there exists an opportunity cost for an approach like mine: if I spend too much time answering impromptu questions and pacing my lecture, I may not be able to cover as much of the course material. I often need to use my judgement to avoid unintentional de-railing of the lecture, and for questions I do not answer in-class due to time constraints, I offer to provide a more detailed answer after the class ends.

ing objectives (e.g., [4]) instead of content to be covered. I have used the backward course design process [3] when serving as instructor of record for both an introductory Computer Science course and also an upper-level elective, and found that having concrete learning objectives and assessment techniques beforehand allows me to better teach the course content (e.g., adequately allocate time to important topics) and effectively assess my students' learning.

Student engagement. The third primary tenet of my teaching philosophy is centered around student engagement. I believe that keeping students actively involved and engaged with the process of learning leads to better course outcomes [2]. Therefore, I structure my lectures to incorporate activities like think-pair-share, in-class demos and coding exercises, and group discussions. Having facilitated learning in a variety of classroom sizes, I have experience with the nuance between employing such activities in a small course (12–15 students) and a large course (150+ students), and my experience teaching has revealed that there is no one-size-fits-all approach to student engagement that works for me. For instance, for smaller classroom settings, I frequently rely on group discussions where students talk within their rows (3–5 per row) before we converse together as a class. By contrast, I find such group discussions impractical for larger classroom settings, and activities like think-pair-share and online or clicker-based polling can be more effective.

structured in-class exercises

I have found that students are often more engaged when the course material is motivated well before being explained. As such, I put an emphasis on *why* we are learning certain topics (e.g., why they might be useful in the future). For a new concept being introduced in class (e.g., a new data structure), I often spend several minutes talking about its applications before even introducing the concept. I have also observed that some students are less likely to participate in-class, due to reasons including imposter syndrome, being non-native speakers of English, personality traits like shyness, among others, and I put an emphasis on reducing barriers to entry for participation for such students. For instance, during lectures, I frequently ask to hear from someone I have not heard from yet, offer to approach students at their seats for direct input before summarizing it to the entire class, and make myself available immediately after a lecture for students to ask individual questions.

facilitating engagement

Teaching Experience

Over the past six years, I have taught in an official capacity to a variety of student groups of varying sizes. I have also participated in several professional development programs to improve my teaching. My teaching experience is summarized below:

- Served as instructor of record for an introductory Computer Science course, an intermediate course, and an upper-level elective.
- Facilitated learning in classrooms of different sizes, including small (10–25), medium (25–60), and large (60–150) settings.
- Designed course materials from scratch and modified existing course content (including developing and tailoring lecture content, assignments, and assessments).
- Helped manage a staff of 30+ teaching assistants (comprising both graduate and undergraduate students).
- Completed the University of Michigan Graduate Teacher Certificate Program, the Computer Science and Engineering Inclusive Teaching Program, and University of Michigan Preparing Future Faculty Program.

Summary

My teaching philosophy is themed around cultivating a growth mindset, administering adaptive lectures, and keeping students engaged in the classroom. I have had an opportunity to develop and demonstrate this philosophy in a variety of classroom sizes and course levels, including teaching three courses as an instructor of record across two institutions. I am interested in teaching at all levels of the Computer Science undergraduate curriculum, and have particular interest in teaching introductory Computer Science with broadening participating in mind.

References

[1] N. Bakula. The benefits of formative assessments for teaching and learning. *Science scope*, 34(1):37–43, 2010.

[2] R. Caceffo, G. Gama, and R. Azevedo. Exploring active learning approaches to computer science classes. In *49th ACM Technical Symposium on Computer Science Education*, pages 922–927, 2018.

[3] Y.-C. Liao and M. Ringler. Backward design: Integrating active learning into undergraduate computer science courses. *Cogent Education*, 10(1):2204055, 2023.

[4] C. W. Starr, B. Manaris, and R. H. Stalvey. Bloom's taxonomy revisited: specifying assessable learning objectives in computer science. *ACM SIGCSE Bulletin*, 40(1):261–265, 2008.