Diversity Statement

Omar Ibrahim

Students don't come to us with a neutral view of computer science and we are often fighting an uphill battle both to attract and retain students in a field that is actively or passively not built for them. It is vital to recognize that much of what we know about teaching computer science is tailored to the existing computer science culture with its diversity and inclusion issues, and may not be best equipped to serve students who fall outside of that culture's stereotypical boundaries.

Building Community and Lowering Barriers

The workshop classes I've taught have been designed to expand participation for underrepresented and minoritized students in CS and support those students through their degrees. These workshops don't just provide academic support; they serve our students by building community and support networks for them. Students from traditionally represented backgrounds frequently either have existing support networks to help them, or they have more resilience for experiencing difficulty because they can look around and *see* that they belong in CS. We have to work to build community for underrepresented students so that they have somewhere to look around and see that *they* belong in CS, too. In my workshop classes, we push the students to work in groups so that they will have people around them for support and to share and normalize common struggles. We also make sure that students have required regular meetings with TAs; it can be hard to admit that you don't know something, particularly when faced with imposter syndrome or stereotype threat. Showing up to office hours voluntarily can feel like an admission of that. By already having to meet with their TAs, students don't have the barrier of voluntarily showing up, and have an easier time discussing what they're struggling with. I also held an imposter syndrome panel in one of my workshop classes; hearing honestly from the people positioned as the experts about experiencing the same struggles shows that having those struggles is okay and normal and not a reason that someone cannot or should not be successful in CS.

Helping Students See Themselves in CS

It is crucial that students can see themselves in the course staff; for underrepresented students especially, it sends a message that people like them are a part of this field and can excel in it. When no one looks like you, it can make you feel like you are not the "norm" in CS. When I started teaching, I noticed MENA (Middle Eastern & North African) students repeatedly requesting to talk to me by name during office hours despite none of them being my students. When I began to see the pattern, I realized they were able to see themselves as a part of the field through me and felt I might be able to understand their experiences in a way that would help them learn. I keep these experiences in mind when I'm involved in hiring so that students can get a broad view of what it can look like to be a person in CS; we ensure that the course staff understands the students' backgrounds and experiences and are better able to serve them, and students have people that they can view as experts and see themselves in.

There are stereotypical ideas of what a person must be to be successful in CS. Designing coursework based on those stereotypes adds extra cognitive load that is distributed unequally

among students; frequently, it puts more cognitive load on underrepresented students to show the same level of understanding. Students shouldn't have to spend time that should be spent learning CS having to learn about DNA or Pokemon just as background to show that they understand CS concepts like loops or objects. In my classes, I work to identify those implicit assumptions and recontextualize those examples using references to background we know students have just by being our students — references to their classes or the university or the city. In my discrete math course, I recontextualized problems about calculus and about video games to discuss the TAs registering for classes or projects that intro students do.

Empowering Student Teachers

I recognize that I was able to end up where I am today because I worked with instructors who mentored and empowered me as a TA. It's how I became interested in teaching as a career, and it's how I've been pushed to become a better teacher by taking courses on Equitable and Inclusive CS Pedagogy and Seeking Educational Equity and Diversity. It's important to me to grow the next generation of TAs, both so that they feel welcome and so that students feel empowered by their instructors. When in a position of authority, I give TAs opportunity and flexibility to contribute in ways that build their confidence and allow them to get a wide variety of teaching experiences. I think it's also important that when I have the power to do so, I advocate for more inclusive teaching practices so that there is a message from the top down that inclusive and equitable pedagogy is an integral part of what it means to teach CS.

Making Accessibility the Default

A vital part of diversity, equity, and inclusion work is accessibility. Accessibility is often something that is managed on an individual basis by a school's disability services, but I think there are ways we can make classrooms more generally accommodating beyond just providing accommodations on an individual basis. Often, accommodations that an individual student might need would just be generally helpful for the overall student population; one example is the student who requested a copy of any slides presented in my class. There's really no reason for me to *not* provide that to all my students, so I can use accommodations in that way to make my classroom more accessible by default, as opposed to accessible upon request. Showing students that accommodations are the default and not the exception helps students to feel more comfortable advocating for their own specific accommodations and perform better. I've been in situations where I did not realize I wasn't properly meeting a student's accommodation, and that can be a scary thing for a student to talk to their instructor about. Showing the importance of accommodation helps students feel comfortable that I will respect and take them seriously, and that lets me grow as an educator and best serve my students. There will always be accommodations that can't be planned for ahead, but I do my best to bake accommodations into the design of my courses.

Cutting Room Floor

- Making assumptions about student backgrounds informed by stereotypes reinforces them and makes students who don't feel that they either do not belong in CS or cannot be successful in CS when in fact, those stereotypes are irrelevant to someone's CS ability.
- We need to make those doors accessible to all, *and* we need to make sure that students are best served once through those doors.

- CS can be difficult and still be for you, but you need to have someone telling you that, otherwise it's easy to say "this was hard, it must not for me".
- An important part of building an inclusive and diverse computer science is reimagining the ways we design classrooms. I've worked to follow best practices around grading for equity to deconstruct what grades actually represent and provide equitable policies to support student learning. This has included shifting classes to use standards
- Working on creating better pedagogical ways to equitably serve our students. Can include things like providing resubmissions, providing opportunities for students to continue learning at their own pace. Not going to inherently learn at same pace. Building systems to acknowledge just because someone is learning at a different pace doesn't mean they are bad at the material, building ways that those students can still be served.
    - Stakes - lowering stakes for students. Ways to equitably handle late work
    - In general try to be flexible about policies. Implementing lateness policies that are not just me being nice, but build in flexibility for students so that they don't have to feel like they need to prove endlessly their reasons for being late.
    - Building systems that are resilient to student circumstances.
- Could also include something about acknowledging who our actual student population we're serving is, and hiring people from related majors who may also frequently be in our classes and have perspectives related to wha those students need to know (i.e. are we assuming the important things to be the things that the majors would find important later vs non-majors)