

## Statement on Justice, Diversity, Inclusivity — Daniel Patterson

The Boston Globe reported in a sweeping series on racism [1] that in 2015 the ten largest private universities in Boston enrolled only around 5% black students, a percentage that for many schools hasn't budged for decades, lower than the Boston area and much lower than the country. In computer science, despite recent efforts, the percentage of women graduating with degrees has *declined* over the decades [2]. This is not a new problem, and it requires deep, structural assessment if we are to address it. One issue is the notion that computer science departments are removed from society, simply teaching a technical skill, and yet outside software has an ever increasing, sometimes not positive, role to play in society: from drones dropping bombs on any male of a certain age [3] to software that systematically denies loans to Black and Latino families [4] to amplifying disinformation on social media.

These issues are tied together, as an apolitical department that takes no stance on the role of technology in society tacitly endorses not only its positive impacts but also its negative impacts, and those negative impacts will disproportionately affect those that are already marginalized in society. Thus, if we want inclusive and diverse departments, we must orient towards the larger goal of having our departments increase justice and reduce inequity in society, not merely to change the face of our departments. This fundamentally means deciding to have the department be a supportive place for those who may not have a background in computer science, may not have had the resources for their own computer or the encouragement that they could succeed in this field. It means ceasing to valorize “college dropouts” like Bill Gates (who learned to program at a fancy prep school in the late 60s) and Mark Zuckerberg (taught to program by his father in middle school), or supposedly self-made man Elon Musk, whose family owned an emerald mine and was programming by age 10. We must realize that these are not people to aspire to, because even if we wished it, these are people that it is effectively impossible for a student to become.

But more, it means reassessing what we do in and outside our classrooms. In one of the introductory classes at Northeastern, which I sat in on and gave a few lectures as a TA, the professor asked an off-hand question at some point during the semester: “How many people know Java?”. The hands went up and down quickly, but at least 85% had indicated they did. What of the 15%? While the class in question was self-selected as an advanced section, I asked if this sort of question could be tracked, if we could do follow up to see if the people who were struggling in the class were the same people who did not have high-school programming experience, as it seemed of fundamental importance: if success in our classes correlates to people who have programmed before, then we will fail exactly those that a commitment to equity should mean we are focused on reaching. We didn't do that then, but this sort of analysis of our outcomes seems critical.

More fundamentally, we must accept that if we enter a classroom hoping to find the brilliant student that will surprise us by how well they understand the material, we will primarily be looking for that student who had earlier opportunities—whether from family members, privileged education, or societal encouragement—and had the time and resources to take advantage of them. I've worked with some of those students—talking to them after class or office hours, or mentoring them on research projects, and while it's undeniably exciting to find someone who grasps the material that you love so deeply, it also runs counter to the entire project that should underlie Diversity, Equity, and Inclusion efforts: to decrease the inequity of society and increase justice.

Some of that, of course, is beyond the scope of a department, or even beyond the scope of a University. But within the department, there is much that we can do—first and foremost, we can

design curricula to cater towards students who do not have a background in computer science. To start, the examples that we draw upon, both for lectures and in assignments, should be both relevant and accessible to someone without prior computer experience. This may, indeed, be hardest for those teaching the class, as we are filled with context of how the introductory material will connect to more advanced material, but if we emphasize those connections, we also emphasize the insularity and alienate those who may be still considering if the field is right for them.

If, instead, from the very beginning we connect the examples we use to the real world, we ground the material and make it more obvious how a student who pursues this can then affect change in that world. Simple programs can model software that they experience in real life, like traffic lights or public transit systems. By bringing in such examples, e.g., doing basic climate simulations or modeling of money using city budgets or wage gaps, we can address pressing issues at the same time as presenting the material. Since we need examples no matter what, this is not a matter of “slowing down” our presentation to account for society, but rather, replacing artificial or abstract examples with those that may connect to the real lives of the people in our classes. If our goal in a curriculum oriented towards justice is to help students figure out how they can reduce inequality in the world, we should, from the beginning, help them understand how the skills that they are learning connect to that world, rather than presenting them with abstract examples divorced from morality and letting them only later realize that their work can have an impact. At that point, they may be either deeply embedded inside huge companies where such exploration is difficult or, perhaps, have dropped out because they didn’t see how they fit in.

We should also strive to attune assessment to what is truly important: what our students are learning. Deadlines should be as flexible as possible to account for the fact that the students that we wish to reach may have families at home, or family members they are taking care of, or jobs outside of school, etc. As a simple example, in a class that I designed and taught, I gave all students a number of no-need-to-ask-me late hours for assignments, with the intention of increasing flexibility in a low friction way. The principle of universal design [6] means that these sorts of interventions benefit everyone.

Beyond course work, we should strive to have our departments do whatever is necessary for students to succeed, and build support for not only student initiatives (clubs, lecture series, etc) but also make more visible the internals of our departments—by holding regular open forums to address diversity initiatives, and use them to seed future work. We should also think of academic advising as a proactive process, where advisors are responsible in flexible ways for their students, not merely as a series of checkboxes that a student needs to satisfy. Where possible, students should be able to choose advisors with backgrounds similar to their own.

These are, of course, but a few examples of interventions we can make, and many more, or different ones, will arise given the particular situation. What is important is having an overall orientation towards justice, being open to making interventions that increase it, and studying whether those interventions are effective.

1. <https://apps.bostonglobe.com/spotlight/boston-racism-image-reality/series/colleges/>

2. <https://nsf.gov/statistics/2018/nsb20181/figures>

3. <https://www.nytimes.com/2012/05/29/world/obamas-leadership-in-war-on-al-qaeda.html>

4. <https://apnews.com/article/lifestyle-technology-business-race-and-ethnicity-racial-injustice-b920d945a6a13db1e1aee44d91475205>

5. <https://www.bostonglobe.com/2021/06/14/metro/city-boston-launch-task-force-review-pilot-program/>

6. Explained, e.g., in “An Inclusive Academy”, Stewart & Valian 2018