DOI:10.1145/3424000

Amy J. Ko et al.

Education

It Is Time for More Critical CS Education

By which 'critical' means an intellectual stance of skepticism, centering the consequences, limitations, and unjust impacts of computing in society.

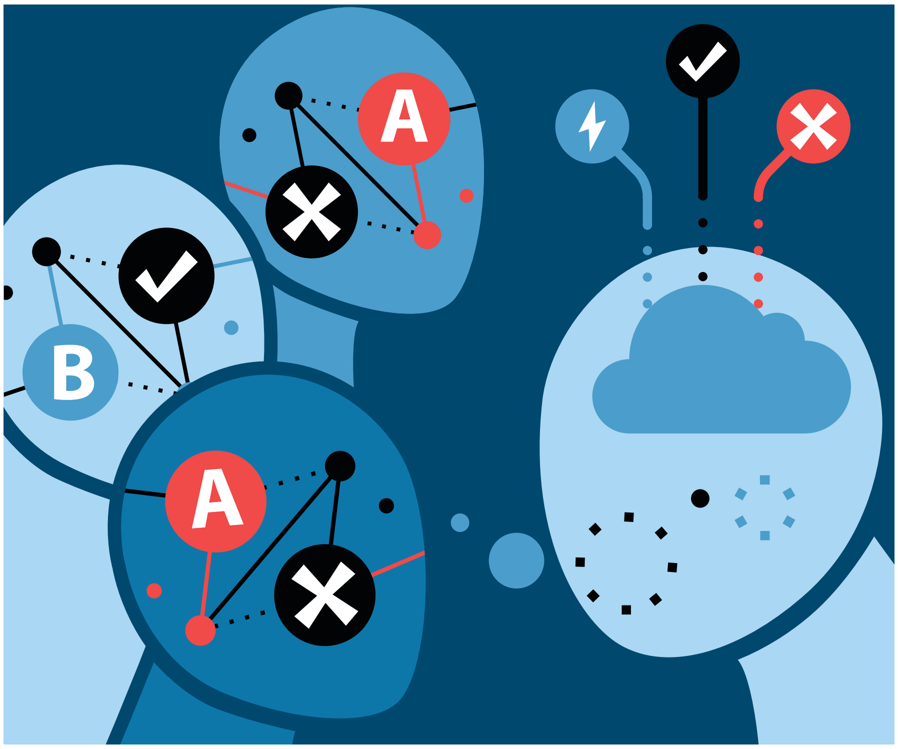

IMAGE BY ANDRIIJ BORYS ASSOCIATES, USING SHUTTERSTOCK

We live in uncertain times. A global pandemic has disrupted our lives. Our broken economies are rapidly restructuring. Climate change looms, disinformation abounds, and war, as ever, hangs over the lives of millions. And at the heart of every global crisis are the chronically underserved, marginalized, oppressed, and persecuted, who are often the first to befall the tragedies of social, economic, environmental, and technological change.[3]

You might think these issues have little to do with computing. But you would be wrong. The weaving of computing through society has not only involved computing in these crises, but, in many ways, placed computing at their centers. Computers increasingly mediate our communication. Automation is accelerating economic restructuring, destabilizing work, and devaluing labor. The demand for information is increasing carbon outputs and exploitative mining of rare metals. Social media is amplifying falsehoods. The Internet is the new battleground of modern warfare. And in all of these systems, data and algorithms amplify racism, sexism, heterosexism, ableism, ageism, xenophobia, cisheteronormativity, and other forms of inequity, injustice, and bias.[2,3] Computing does not occur in a vacuum: it shapes and is shaped by ever-evolving social, cultural, institutional, and political forces.

These links between computing and injustice seem invisible to many, including those who bear the brunt of these injustices. Some young people grow up seeing computers as magical machines that bring joy, escape, and connection. Others experience them as vectors for violence, sexual harassment, cyberbullying, addiction, and isolation. Some adults view computing as a force of economic growth and progress. Others experience subjugation to unjust algorithmic decisions about their loan eligibility, work schedules, and unemployment insurance, yet lack the computing literacy to counter authoritative voices on these algorithms' designs. Meanwhile, many of us in the computing discipline, while happy to celebrate computing as a tool for social change,[1] ignore its role in these injustices,[2] and in some cases, dismiss the idea that computing is anything but a value-neutral tool independent from society.

We argue, as others have,[5] that making these injustices visible to society is the responsibility of CS educators.

After all, educators hold the power to shape public perception of computing. We do this through the problems we focus on in our classrooms; through who we choose to teach; in how we shape students' career choices; and in how we conceptualize computing to journalists, social scientists, and society. The world has critical questions about computing and it is time we started teaching more critical answers.

While there are many ideas to teach, we believe three ideas are key.

Computing Has Limits

Computing is powerful and the allure of this power is compelling. It is what drives students to our classrooms, it is what has led to worldwide calls for CS for All in primary and secondary schools, and it is what has made some of our lives better than ever, providing more information, connection, opportunity, and voice.

But the belief in computing's limitless power has led many of us to believe that computing *always* makes things better.[1] This could not be further from the truth. Judges, for example, have begun to delegate sentencing decisions to recidivism prediction software, ignorant of the racially biased data upon which those predictions are based. Our global climate agreements rest heavily upon the assumption that technology, and not behavior change, will save us from calamity. Investors have amplified the computing-enabled gig economy not because it is an inherently more humane form of human labor, but because it profits a small group of private investors and saves those with means and money a bit of time.

All of these troubling trends emerge from a set of neophilic myths: that software is always right, that software is always value-neutral, and that software can solve every problem. CS education must replace these conceptions with the reality that software is often wrong; software always embeds its creators' values and biases; and software can only solve some problems, and many cases, creates new ones.

Data Has Limits

Computing has little value without data. People come to Facebook not for the newsfeed algorithm, but for the content their friends and family write. People come to Google, Baidu, and Yandex not for ranking algorithms, but for the Web pages millions have carefully authored. People watch Netflix, iQIYI, and Tencent not for their recommendations, but for television, movies, and events. And while these algorithms are useful, their value is dependent on the quality of the data they process: imperfect, biased inputs lead to imperfect, biased outputs.[3]

But computing often subordinates data, ignoring the cost of creating it, the individuals and social contexts from which it is wrought, and its role in global crises and injustices. After all, it is the desire for data that drives the carbon output of datacenters; it is biased datasets that enable facial recognition algorithms to work so well for white people, subjecting everyone else to greater risk of accidental prosecution by automated surveillance; and it is binary classifications in airport security scanners that, trained on cisnormative bodies, cause trans and non-binary people to be physically harassed for "bodily anomalies."[2] Data is responsible for many harms of computing, whether directly through its collection or indirectly through its use.

Thus, all CS educators must teach what information science and librarians have long known: data is always about the past and not the future; data is always an imperfect and biased record, encoding the values, beliefs, and ideas of its creators; and incorrect interpretations and uses of data harm people in unequal ways.[4]

Data is responsible for many harms of computing, whether directly through its collection or indirectly through its use.

CS Has Responsibility

Many early conceptions of CS education view computing as a medium for expression. And this view has dominated: we celebrate what students and companies create, partly in recognition of the inherent difficulty of programming. But while we give great attention to how our students create things, and the scale of impact their creations have on the world, we often leave the moral choice about what to create to individuals and investors.

However, the choices that developers make when they create with computing are not purely individual or capitalistic. They are inherently social and collective, and infused with value judgments. For example, when a CS graduate accepts their first job, they are endorsing and investing in the values of the company they choose; students should be supported in reflecting on this endorsement. Similarly, when engineers at Google internally protested the creation of a censored search engine for China, they were doing it on behalf of not only themselves, but China and the rest of the world.

CS education at all levels must center these responsibilities and value tensions, ensuring all people—not just CS majors—understand that creating software comes with collective responsibilities to society.

Ways Forward

Many respond to these concerns by advocating for everyone to learn to code, arguing that programming forces us to confront the limitations of computing, the necessity of data, and the role of programmers in shaping software. But learning to code often leads people to view programs as powerful rather than perilous, data as abstract and free of bias, and programmers as clever wizards rather than social actors. And yet, more people know how to code than ever, and critical views on computing are still rare in CS education and industry.

What will make them more common? An intentional effort to develop a critical literacy of computing, helping everyone understand the social and cultural systems that drive computing, and the social and cultural systems disrupted by computing. This means educating primary, secondary, and post-secondary CS teachers who can help everyone see computing as both a powerful medium for expression *and* a perilous tool for oppression. It means preparing CS teachers who can develop students' sense of collective civic responsibility. And it means more than just an ethics requirement for CS majors: it means recasting computing itself in moral, ethical, and social terms.

Realizing a more critical CS education requires more than just teachers: it also requires CS education research.

Realizing a more critical CS education requires more than just teachers: it also requires CS education research. How do we teach the limits of computing in a way that transfers to workplaces? How can we convince students they are responsible for what they create? How can we make visible the immense power and potential for data harm, when at first glance it appears to be so inert? How can education create pathways to organizations that meaningfully prioritize social good in the face of rising salaries at companies that do not? And how do we prepare outstanding primary, secondary, and post-secondary teachers to equitably teach these ideas to everyone in a way that is responsive to local needs and values?

If we can answer these research questions and enact their implications in our teaching, we may see students create (and demand) a more inclusive future for computing. We may see social media stabilize free press and democracy rather than supplant it. We may see a generation of students choose to invest their skills in broader global problems of healthcare, energy, education, and government. And we might see a more just use of algorithms and machine learning.

Work on these futures has only just begun. Researchers around the world are shifting their attention to algorithmic fairness, data bias, and CS ethics education. Grassroots communities are advancing design justice, critically analyzing the role of computing in society.[2] Even ACM's own Future of Computing Academy, which periodically brings together new computing faculty to envision the discipline, recently called not for more innovation, influence, or impact in CS, but more humility. These grassroots movements outside of computing, and our own nascent conversations within computing, inspire some hope. Now it is time to translate that hope into more critical CS education. C

References

1. Ames, M.G. *The Charisma Machine: The Life, Death, and Legacy of One Laptop per Child.* MIT Press, 2019.
2. Costanza-Chock, S. *Design Justice: Community-led Practices to Build the Worlds We Need.* MIT Press, 2020.
3. O'Neil, C. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy.* Broadway Books, 2016.
4. Rubin, A. Learning to reason with data: How did we get here and what do we know? *Journal of the Learning Sciences 29*, 1 (2020).
5. Vakil, S. Ethics, identity, and political vision: Toward a justice-centered approach to equity in computer science education. *Harvard Educational Review 88*, 1 (2018).

Amy J. Ko (ajko@uw.edu) is a professor in The Information School, University of Washington, Seattle, WA, USA.

Alannah Oleson (olesona@uw.edu) is a Ph.D. student in The Information School, University of Washington, Seattle, WA, USA.

Neil Ryan (neilryan@cs.washington.edu) is a Ph.D. student in The Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, WA, USA.

Yim Register (yreg@uw.edu) is a Ph.D. student in The Information School, University of Washington, Seattle, WA, USA.

Benjamin Xie (bxie@uw.edu) is a Ph.D. student in The Information School, University of Washington, Seattle, WA, USA.

Mina Tari (minatari@uw.edu) is a Ph.D. student in The Information School, University of Washington, Seattle, WA, USA.

Matthew Davidson (mattjd@uw.edu) is a Ph.D. student in The College of Education, University of Washington, Seattle, WA, USA.

Stefania Druga (st3f@uw.edu) is a Ph.D. student in The Information School, University of Washington, Seattle, WA, USA.

Dastyni Loksa (dloksa@towson.edu) is an assistant professor at Towson University, Towson, MD, USA.

This column is derived from the first author's 2019 keynote at the ACM Koli Calling conference, available at http://faculty.uw.edu/ajko/talks