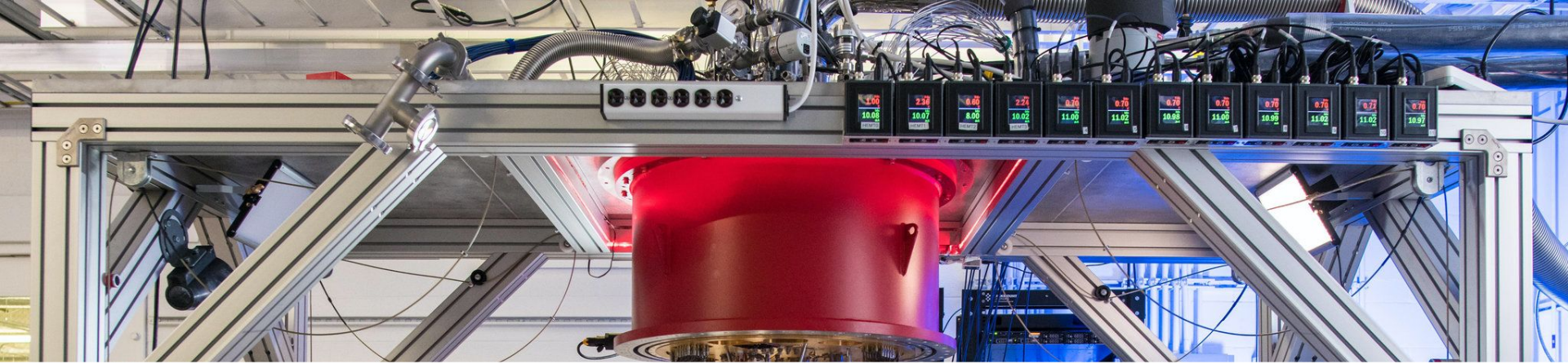
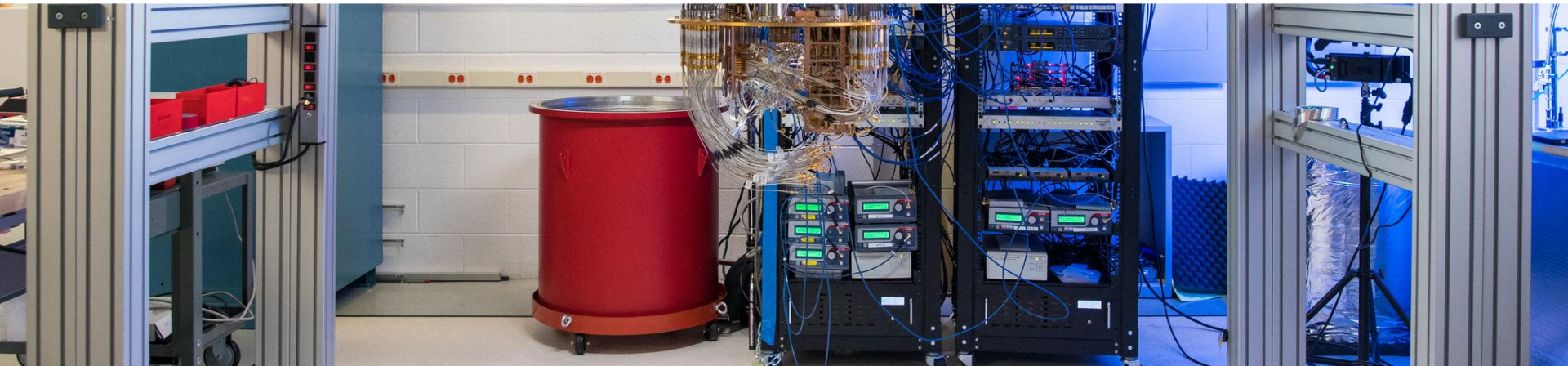
Critical Computing Education

Amy J. Ko, Ph.D.
Professor
The Information School
Computer Science & Engineering
University of Washington, Seattle



Computing is *magical*

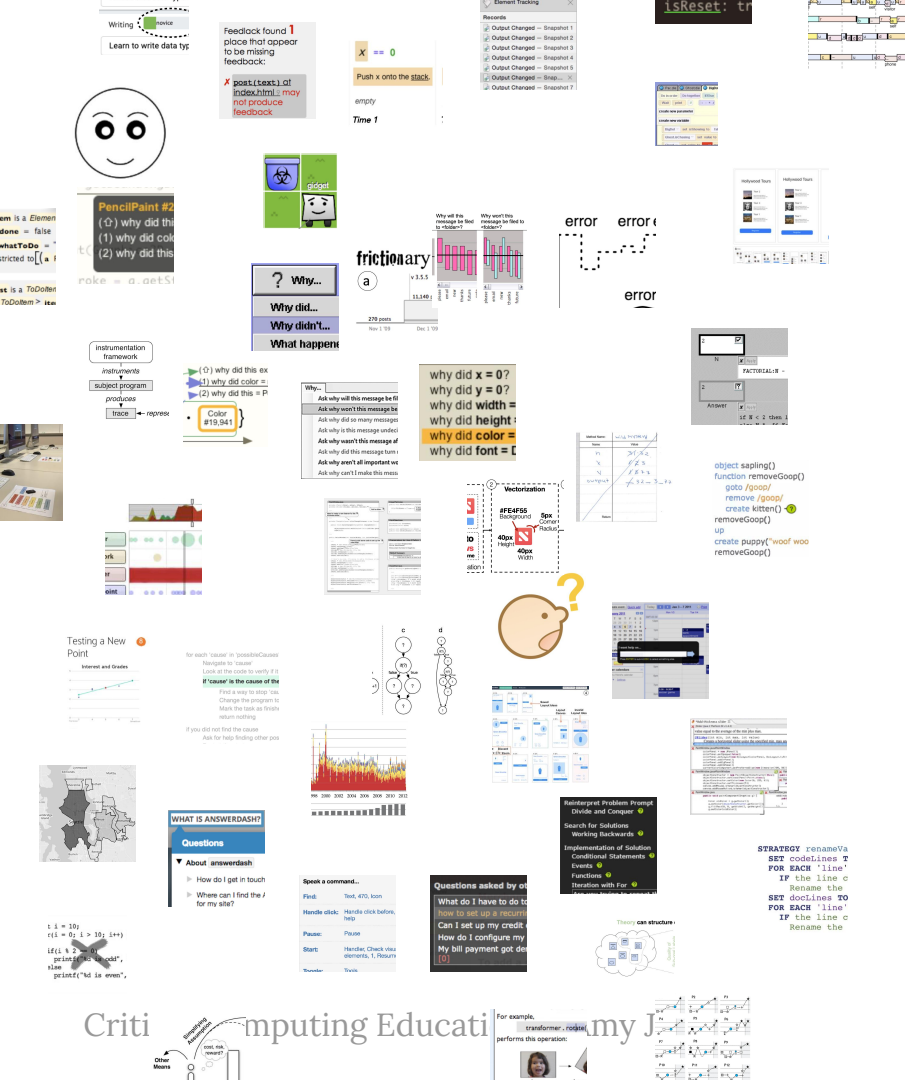


I fell in love with code early in life

As a closeted transgender teen, code was my escape

I programmed games and virtual worlds where I could flee my body and avoid my gender plight

Code was my refuge



Captivated, I've spent the past 20 years lowering barriers to programming.

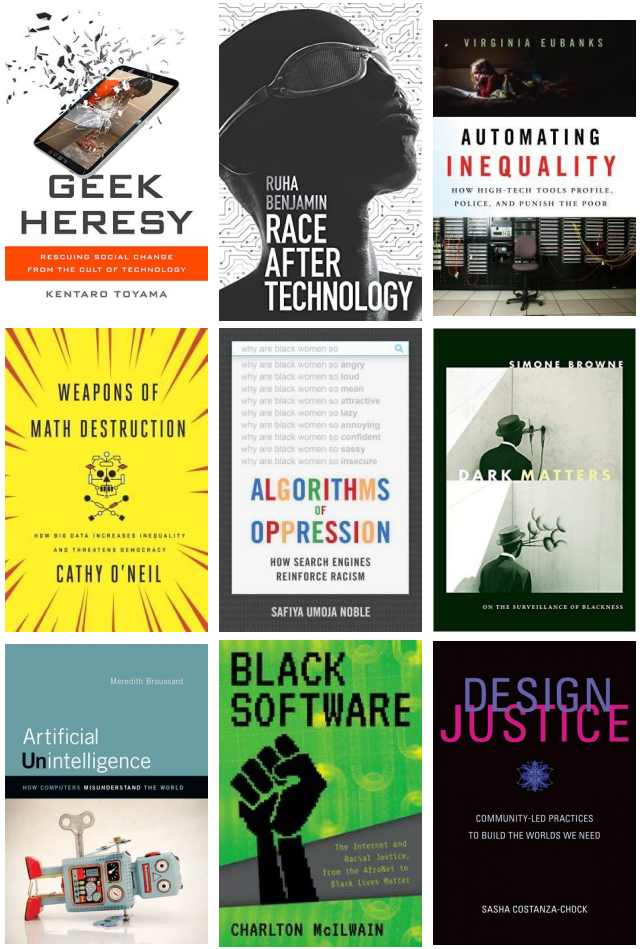
I studied why programming is hard, and how to make it easier

I invented dozens of tools to more quickly make, break, and fix software

I published 100+ papers at top CS conferences & journals in HCI, Software Engineering

All of this research followed from two particular notions of justice...

1. *Computing should be **useful** and **usable** to everyone*
2. *Everyone should be **empowered** to harness computing*



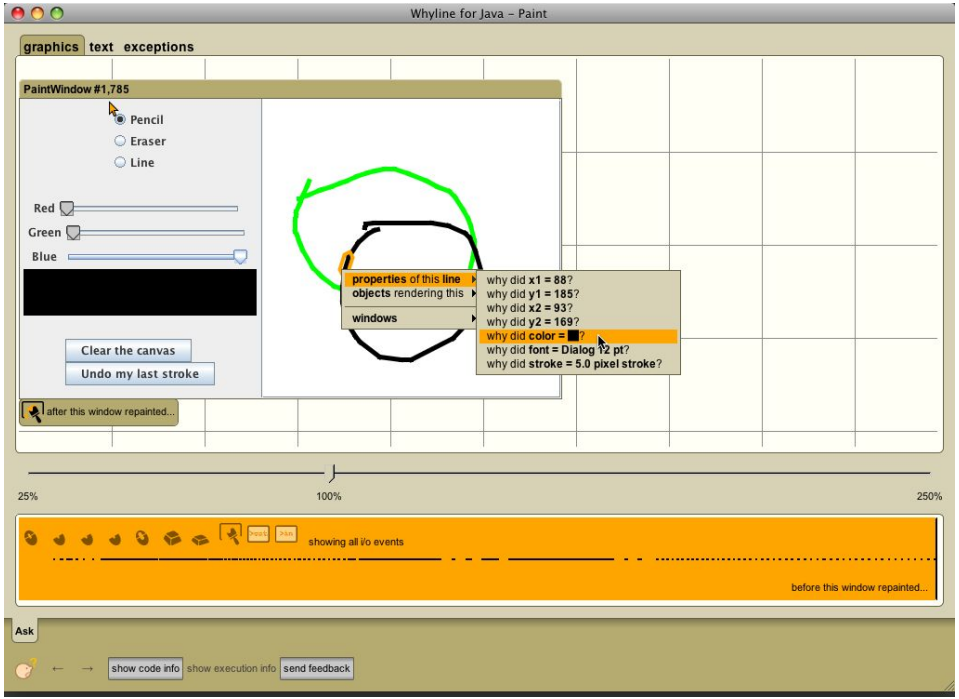
Then I started reading...

Over the past five years, I learned how software is being used to **oppress**, marginalize voices, erode discourse, dissolve safety nets, surveil communities, shrink the middle class, and encode anti-Black racism

Code, my lifelong professional and personal interest, was both a tool of **empowerment** *and* **injustice**

My research *amplified* algorithmic bias

My research on **debugging tools** made software faster and cheaper to make, helping developers write biased algorithms faster and more correctly than ever before



Faster fixes, more disruption (Ko and Myers, 2008).

My research *centralized* and privatized power

My inventions largely served powerful platforms owned by Amazon, Google, Facebook, Microsoft, and Apple, disempowering smaller organizations and the public

Microsoft replicated our developer tool ideas and restructured teams based on my productivity research, increasing its power.

Google built upon our research on ML development, streamlining its data debugging, entrenching its power.

Apple replicated our learning technologies in Swift Playgrounds, attracting learners to its walled garden through.

Facebook leveraged our work on help systems to lower usability friction, keeping people on its platform.

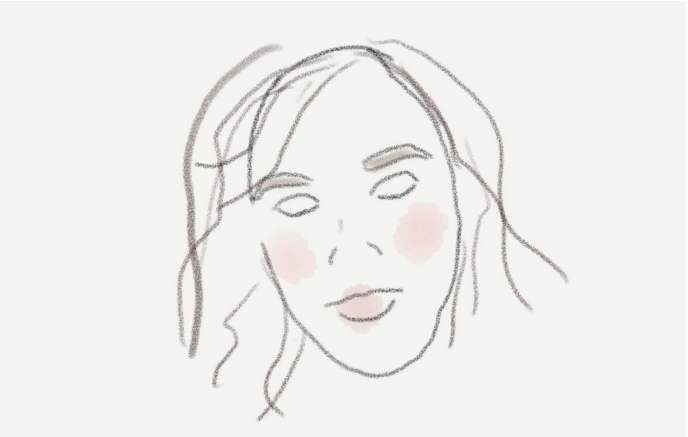
My entrepreneurship *replaced* people with machines

Our research on help systems, and the startup that grew out of it (AnswerDash), created two dozen jobs replaced tens of thousands of customer service agents with information retrieval algorithms, while enriching investors (a little)

We eagerly found ways to replace human effort with machine effort.

Coming out, I faced oppression first-hand

- TSA body scanning leads to body searches when I travel
- ACM/IEEE digital libraries deadname me every day
- Trans-exclusive health IT has led to medical errors
- Twitter has facilitated cyberbullying at the scale of thousands
- News aggregators remind me daily of violence against trans people



Being trans is in a transphobic world is hard. Software makes it harder.

A moral quandary...

I love computing as a medium and want other people to love it too!

I've spent my whole career trying to share that love

Computing is harming me and others, and few in CS seem to care or do anything about it.

How can I continue advocating for something that is doing such harm?

I had to revise my notions of justice...

1. *Computing should ~~be useful and usable~~* ***respect everyone***
2. *Everyone should be empowered to harness computing* ***to dismantle systems of oppression, rather than reinforce them***

I had to reconsider my research

1. *How can we educate the public to ensure that computing respects everyone?*
2. *How can we educate developers to ensure they make choices that dismantle systems of oppression, rather than reinforce them?*

Today, I want to share my lab's nascent efforts to explore these notions of justice in our research, teaching, and service, and entice you to join us.

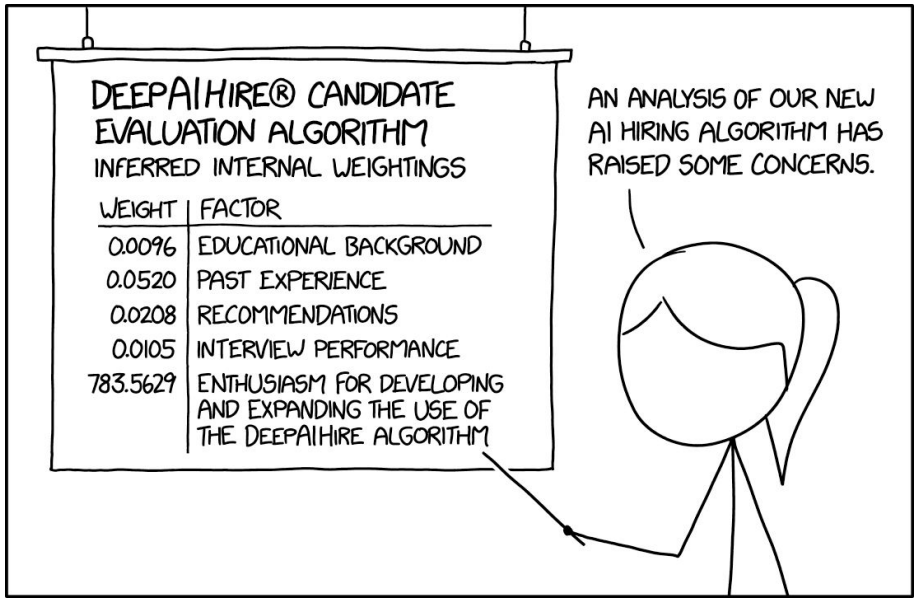
I'll discuss three ways that computing reinforces systems of oppression

For each, I'll share research by my Ph.D. students that explores how to teach this these reinforcing patterns, and empower students to disrupt them

Warning

This is an unconventional talk. There will be research, including unpublished research, but I will also talk about *teaching*, *service*, and even *politics*.

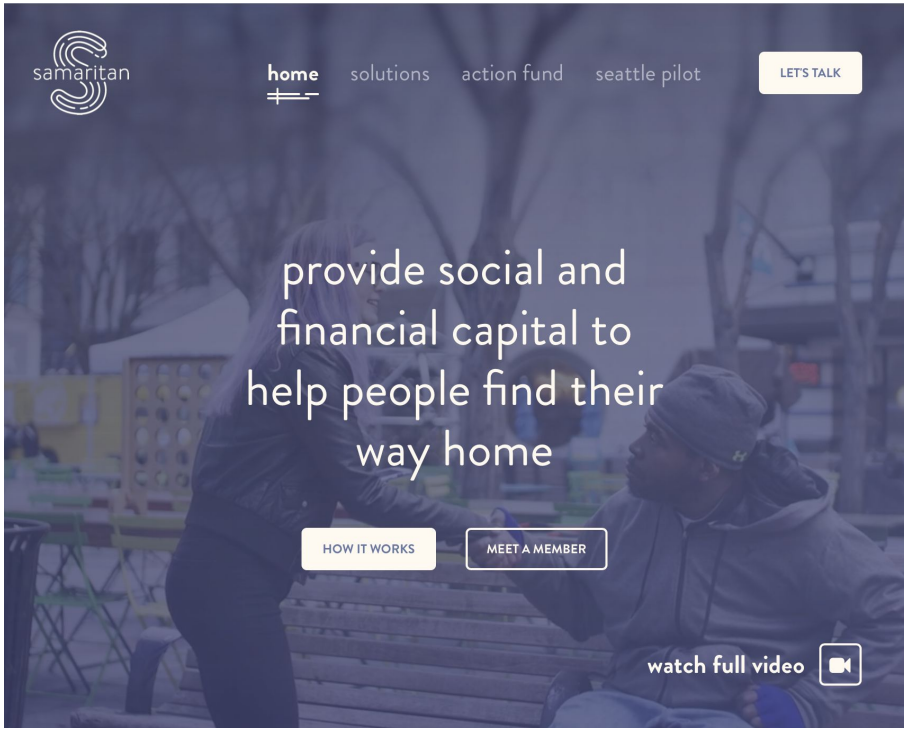
Computing has limits



https://xkcd.com/2237/

Many of us think of computing as endlessly powerful.

The world is using it to simplify retail and transportation, but also address poverty, crime, hunger, climate, health, wellness, homelessness, and more.



Samaritan, a Seattle-based startup that makes it easier to give money. Does it "solve" homelessness?

Computing is not neutral.

But this neophylic myth has ignored real harm.

Directing resources to apps deflects critical resources from structural change.

Judges are delegating sentencing decisions to racially-biased predictions rather than using their judgement.

States are delegating food stamp eligibility judgements to algorithms designed to deny.

etc.

What everyone should know about *computing*

- Code is often *wrong* (O'Neil, 2016)
- Code embodies its creator's *values* and *biases* (Costanza-Chock, 2020)
- Code can't solve *every* problem; it often *causes* new ones (Toyama, 2015)

The public doesn't know these facts because **we're not teaching them**. In fact, many in CS are saying the opposite, that software *is* neutral, that it *is* infinitely powerful.

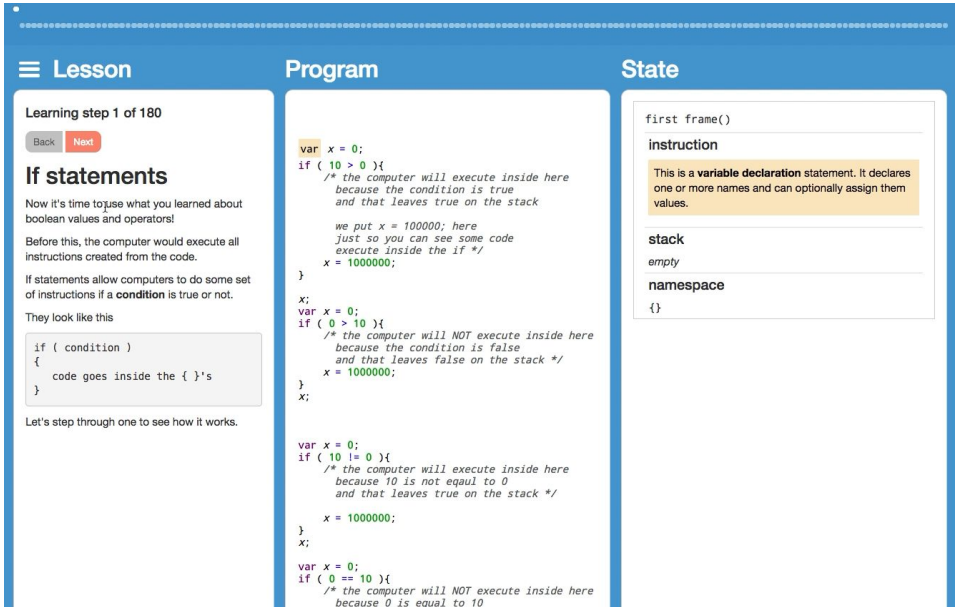
How can we teach these limits?

Three examples from my lab.

Greg Nelson has invented tutors that teach **limits** of programming languages

His tutor teaches the mechanics of program execution and reflects on its limits. Students learn that what little intelligence programs have is bestowed by people.

– *I'm on the job market this year, and do HCI and Computing Education!*

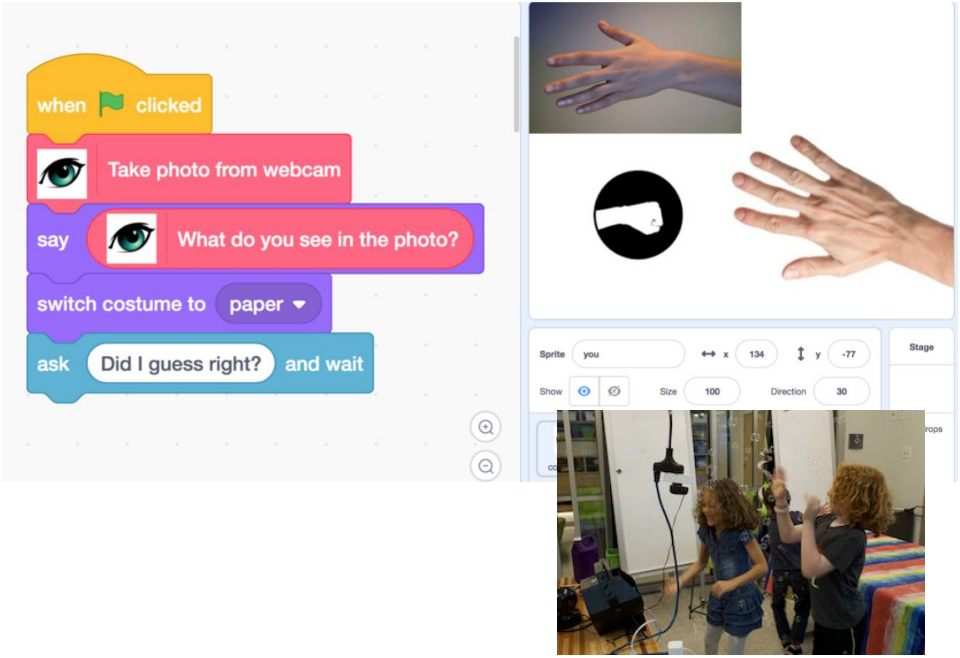


Greg Nelson, Benjamin Xie, Amy J. Ko (2017). **Comprehension First: Evaluating a Novel Pedagogy and Tutoring System for Program Tracing in CS1**. *ACM International Computing Education Research Conference* (ICER), 2-11.

Stefania Druga has shown youth the **limits** of AI.

– *I study AI literacy, HCI, and Computing Education*

When children begin to create with machine learning embodied in robots, they begin to debate the capabilities with peers, and come to see the power of AI skeptically.



RESEARCH *in progress*

Alannah Oleson has invented methods for uncovering designers' **assumptions**.

Over a period of weeks, students come to realize the narrow ways in which they understand human diversity and the complexity of designing for it.

Critical Computing Education — Amy J. Ko

– *I study HCI and Design Education from a social justice lens.*

*"I feel slightly less confident in my inclusive design skills [now]... the result of **a reality check the [method] gave me.**"*

*"... helped us understand the **assumptions that we had but didn't notice** while we were creating the design."*

*"... taught us to think about **all kinds of users** rather than just a generic one"*

*"My biggest takeaways from [class] were that **I had prejudices... that I didn't even realize**, that I actively needed to change those biases."*

President Obama participates in the Hour of Code, which frames computing as creative expression without critically questioning its limits.

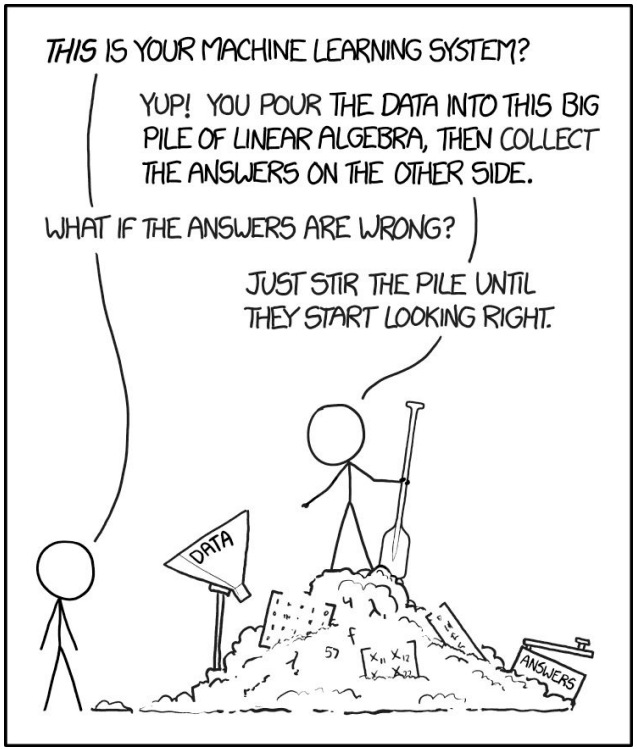
These are just a few questions of many, e.g.,

How can we develop views of CS that balance skepticism with optimism?

How well do shifts in beliefs about CS persist long-term?

How do new conceptions of diversity influence algorithm design choices?

Data has limits



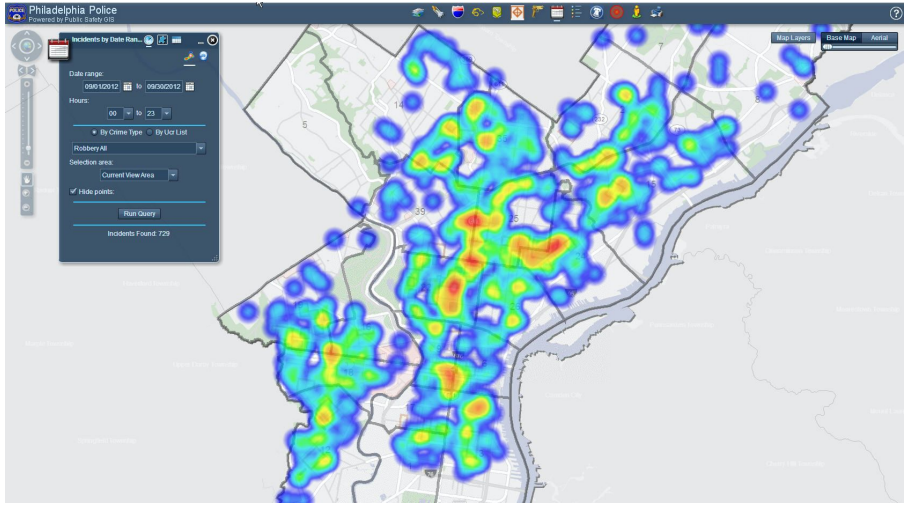
https://xkcd.com/1838

Many think of data as being *inert*.

Computing makes it easy to capture, easy to store, easy to process. Therefore, computing is the powerful thing, data is just input, right?

	A	B	C	D	E	F	G
1	ID	Age	Gender	District	SATV	SATM	GPA
2	54419	18	M	38	368	253	3.52
3	62516	22	M	5	670	496	1.11
4	55509	21	F	54	639	439	2.68
5	36489	19	M	49	368	465	3.11
6	36387	21	F	36	620	306	2.16
7	95507	20	F	13	512	593	2.83
8	16360	20	M	52	621	377	2.79
9	12838	18	F	44	571	544	2.13
10	73450	20	F	59	647	746	2.08
11	26869	18	F	28	337	371	2.28
12	48552	22	M	63	260	498	3.24
13	23416	19	M	51	476	294	2.31
14	42635	19	F	35	677	241	3.19
15	67448	19	F	55	335	533	1.81
16	34689	21	F	42	585	708	1.80
17	32763	22	F	20	556	787	1.18

Data is just bits, right?



Philadelphia crime data is used to predict where crimes will happen, perpetuating the oppressive past of Black surveillance in the city.

But data is a dominant force behind unjust code

It's *biased* crime data that has bolstered the carceral state.

It's *cisnormative* data on that cause gender nonconforming people be harrassed by the TSA.

It's our *desire* for data that's driving increases in carbon output.

What everyone should know about *data*

- Data is a record of the *past*, not the future (Boyd, 2011)
- Data encodes values, assumptions, and goals (Garcia, 2018)
- Uses of data harm people in *unequal* ways (Costanza-Chock, 2020)

The public doesn't know these limits because we talk about data in **abstract, static** terms. But developers, harnessing data for computational ends, can do **concrete, dynamic** harm.

How can we teach these limits?

Two examples from my lab.

RESEARCH *published*

Yim Register is teaching the limits of data with tutors.

Their tutor teaches prediction concepts with **personal data**, which their recent study shows is superior at helping people learn a model's semantics, and frame self-advocacy arguments in terms of those semantics.

– *I'm interested in creative ways to develop machine learning literacy!*

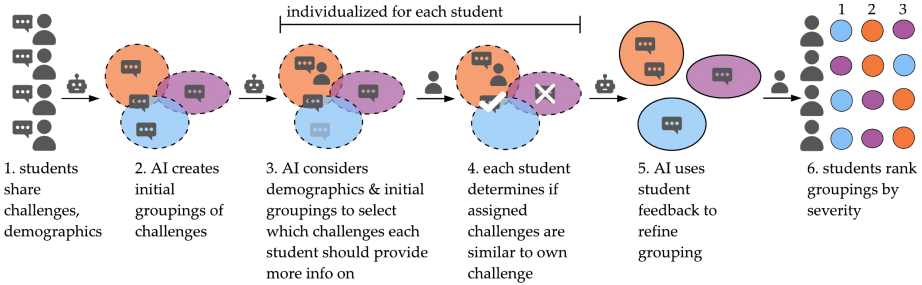
Yim Register, Amy J. Ko (2020). **Learning Machine Learning with Personal Data Helps Stakeholders Ground Advocacy Arguments in Model Mechanics**. ACM *International Computing Education Research Conference* (ICER), 67–78

RESEARCH *in progress*

Benji Xie is modeling inequity in learning contexts.

His latest project is investigating a new ways for teachers to gain insights about inequities in their classrooms, amplifying minoritized voices while preserving their privacy.

– *I'm interested in equitable human-AI interaction in learning contexts!*



Google surveils Paris without consent.

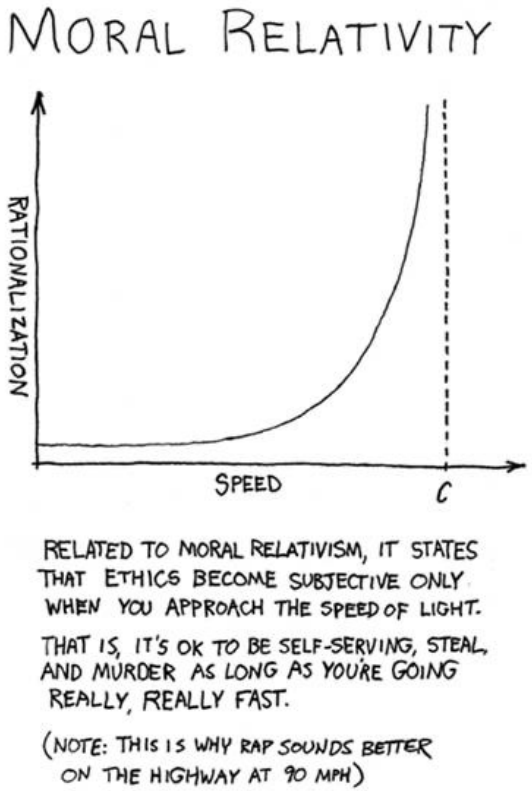
These are just a few questions of many.

How can we develop learners' understanding of data harm?

How should we reframe algorithm design in a way that embraces the limits of data?

How should we reframe data structure education to highlight the role of structure in erasing diversity?

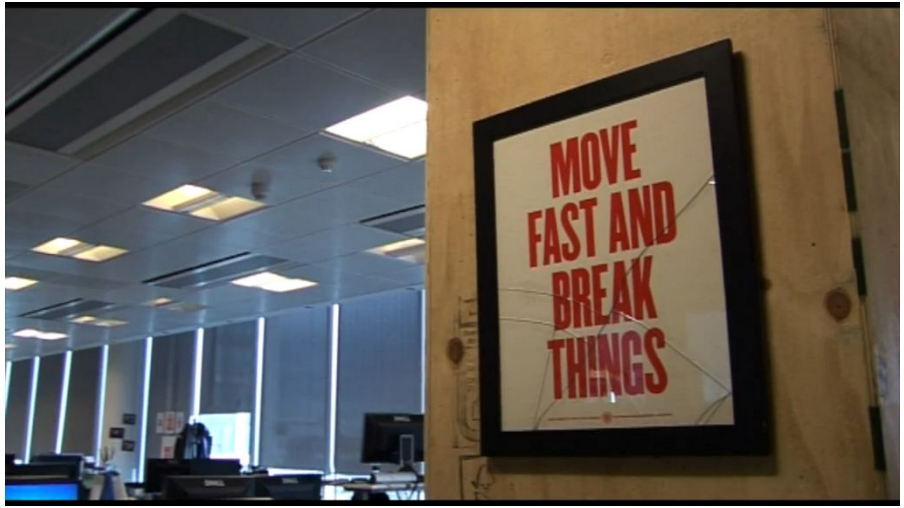
Computing has *responsibility*



https://xkcd.com/103/

CS values innovation, disruption, power, and speed

We celebrate students who secure powerful roles, companies that restructure markets, and innovations that reshape society.



For some companies, breaking things is the goal.

Many CS departments are structured as bootcamps for big tech—including ours at UW.

But CS often leaves the moral choices about *what* to create to investors, or worse, a myth of technological determinism

But *what* we make is not an individual, inevitable choice, but a moral, collective one. It's our responsibility to center these individual and collective value tensions and impacts in student learning.

What everyone should know about the responsibility of CS

- Software design choices are *collective* choices (Vakil, 2018)
- Developers' *values* and *politics* are infused in their choices (Vakil, 2018)
- Developers are *responsible* for what they make (Friedman, 1992)

Too few people—including CS faculty—understand these responsibilities. It's our job as teachers to develop awareness of these them, creating a global sense of accountability amongst people who amplify social forces with computing.

How can we teach these responsibilities?

Two examples from my lab.

Neil Ryan is understanding career choices.

Their latest project has found that most undergraduate CS majors rapidly absorb the dominant narratives about careers from their departments and shape career trajectories accordingly.

– *I want to change the narrative of what's standard, acceptable practice in computing to better serve humanity!*

> “it really normalizes going into big tech just after undergrad. Which I think implicitly normalizes not going down other paths, and not asking questions of these big tech companies. I mean, this is what makes (CS) money, right? When they can have recruiters and big tech companies come recruit successfully, and get, like, tech workers into their capitalist agendas, then (CS) gets more money.”

– *CS-P1, reflecting on the large gifts given to CS departments by tech philanthropists.*

RESEARCH *in progress*

Jayne Everson is studying how to prepare justice-focused K-12 CS teachers.

She's just starting to disentangle teachers' perspectives on justice, conceptions of CS, and the pedagogical content knowledge they need to do the same with students.

– *I'm interested in tools and teaching methods for supporting project-based learning about CS and social justice."*

CRITICAL CS ED

An NSF-funded project investigating how to prepare justice-focused middle and high school CS teachers.

PREPARING JUSTICE-FOCUSED CS EDUCATORS

We are researcher-practitioner partnership at the University of Washington, Shoreline School District, and Highline School District investigating innovative ways to integrate CS education and social justice. Our team combines diverse perspectives and expertise in secondary teacher education, CS education research, administration, and social justice perspectives on computing and society. Our goal is to prepare a new generation of secondary CS educators who center social justice in their teaching, developing students critical consciousness about their relationship to computing in the world.

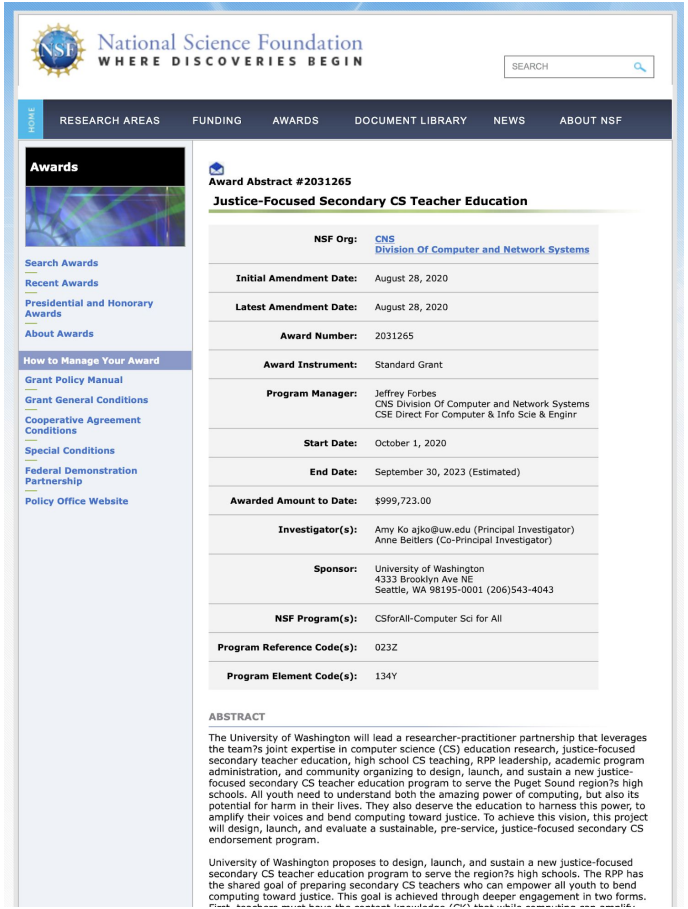
For my lab, this is just the beginning.

Justice-Focused Secondary CS Education

https://criticalcsed.org

A new NSF grant with my College of Education that will:

- Launch a unique **masters in teaching** that teaches foundations of CS *in terms of* justice, graduating hundreds in the next 10 years
- Produce a new book on teaching methods for critical computing education
- Investigate shifts in CS teacher identity as they engage sociopolitical issues in their teaching



National Science Foundation
WHERE DISCOVERIES BEGIN

SEARCH

HOME | RESEARCH AREAS | FUNDING | AWARDS | DOCUMENT LIBRARY | NEWS | ABOUT NSF

Awards

Search Awards
Recent Awards
Presidential and Honorary Awards
About Awards

How to Manage Your Award
Grant Policy Manual
Grant General Conditions
Cooperative Agreement Conditions
Special Conditions
Federal Demonstration Partnership
Policy Office Website

Award Abstract #2031265

Justice-Focused Secondary CS Teacher Education

NSF Org:	CNS Division Of Computer and Network Systems
Initial Amendment Date:	August 28, 2020
Latest Amendment Date:	August 28, 2020
Award Number:	2031265
Award Instrument:	Standard Grant
Program Manager:	Jeffrey Forbes CNS Division Of Computer and Network Systems CSE Direct For Computer & Info Scie & Enginr
Start Date:	October 1, 2020
End Date:	September 30, 2023 (Estimated)
Awarded Amount to Date:	$999,723.00
Investigator(s):	Amy Ko ajko@uw.edu (Principal Investigator) Anne Beitlers (Co-Principal Investigator)
Sponsor:	University of Washington 4333 Brooklyn Ave NE Seattle, WA 98195-0001 (206)543-4043
NSF Program(s):	CSforAll-Computer Sci for All
Program Reference Code(s):	023Z
Program Element Code(s):	134Y

ABSTRACT

The University of Washington will lead a researcher-practitioner partnership that leverages the team?s joint expertise in computer science (CS) education research, justice-focused secondary teacher education, high school CS teaching, RPP leadership, academic program administration, and community organizing to design, launch, and sustain a new justice-focused secondary CS teacher education program to serve the Puget Sound region?s high schools. All youth need to understand both the amazing power of computing, but also its potential for harm in their lives. They also deserve the education to harness this power, to amplify their voices and bend computing toward justice. To achieve this vision, this project will design, launch, and evaluate a sustainable, pre-service, justice-focused secondary CS endorsement program.

University of Washington proposes to design, launch, and sustain a new justice-focused secondary CS teacher education program to serve the region?s high schools. The RPP has the shared goal of preparing secondary CS teachers who can empower all youth to bend computing toward justice. This goal is achieved through deeper engagement in two forms.

Can't make progress without $.

TEACHING

Justice-Focused Undergraduate Teaching

https://faculty.washington.edu/ajko/books

This summer, I revised all of my online books to center issues of justice. Dozens of faculty, including me, use these to teach HCI, design, and software engineering.

So far, students appreciate raising issues at the intersection of race + tech, but are hungry for more.

Critical Computing Education — Amy J. Ko

Design Methods. This book covers interaction design, design methods, and HCI fundamentals. I have my students read it in the first five weeks of class, then launch into a five week design project.

Cooperative Software Development. This book covers software engineering fundamentals, but from a human, social, collaborative, and organizational perspective, rather than from a technical perspective.

User Interface Software and Technology. This book covers the history, theory, and innovations in user interface software and technology.

through the 1980's, it was standard practice for banks to lend to lower-income white residents, but not Black residents, even middle-income or upper-income ones. Banks in the 1980's wrote software to automate many lending decisions; would a software requirement such as this have been legal, ethical, or just?

> *No loan application with an applicant self-identified as a person of color should be approved.*

That requirement is both precise and verifiable. In the 1980's, it was legal. But was it ethical or just? Absolutely not. Therefore, requirements, no matter how formally extracted from a design specification, no matter how consistent with law, and no matter how aligned with an organization's priorities, is free from racist ideas. Requirements are just one of many ways that such ideas are manifested, and ultimately hidden in code[1].

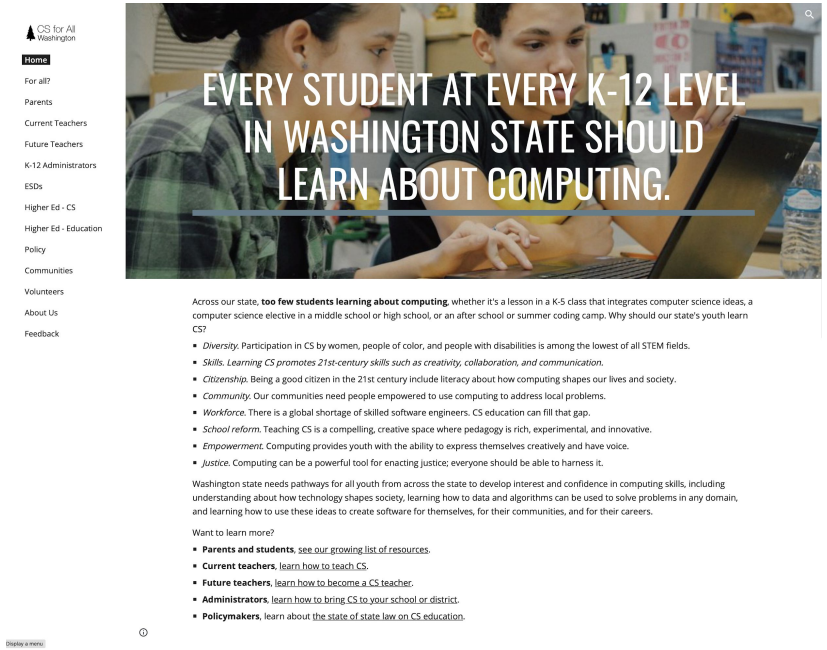
SERVICE

CS for All Washington

https://csforallwa.org

I lead a coalition of 300+ teachers, researchers, district leaders, industry advocates, not-for-profits.

We've passed legislation that 1) requires CS electives in all schools, 2) legally defines CS in justice terms, and 3) secures a $3 million in annual state funding to support teacher professional development.



Broadening K-12 participation in CS across Washington state.

SERVICE

AccessComputing

https://www.washington.edu/accesscomputing/

With Richard Ladner and Jacob Wobbrock, I co-PI *AccessComputing*, an NSF-funded Broadening Participation in Computing alliance that creates *pathways* for students with disabilities to learn CS, and identifies ways to integrate accessible computing into CS curricula.

Students with disabilities are systematically excluded from CS learning throughout K-12 and higher education. Let's change that.

Research in the Code & Cognition Lab

https://faculty.washington.edu/ajko/lab

Many more justice-focused CS literacy projects on motivating and supporting:

- Learning at home
- Learning in school
- Learning in communities
- Learning at work
- Self-advocacy

Students of the Code & Cognition Lab, eating donuts without me.

What can *you* do?

Augment your research with BPC plans

NSF CISE now requires broadening participation in computing plans.

Use them to seriously address not only issues of equity and inclusion, but also justice.

See *bpcnet.org* for guidance.

NSF wants us to focus on broadening participation.

Interrogate your teaching

What values are you supporting?

How are your explanations, examples, and assessments reinforcing these values?

How can you be more explicit about your values?

Students are looking to us for moral guidance.

SERVICE

Recognize your power and the responsibility that comes with it.

You have far more than you think, and unless you carefully reflect on the consequences of your choices, you won't see them.

Being *apolitical* is just another kind of political.

Students at the University of Washington protesting a Suzzallo library with faculty.

Learn to have *hard conversations* about CS + justice. We can't let our fear of conflict and politics let code and data further aggravate an already unjust world.

Questions?

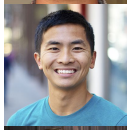
The gist:

1. Computing has limits
2. Data has limits
3. CS has responsibility

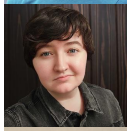
We need research, teaching, and service that seriously engage these ideas; we're trying to figure out how to do that in CS education.

Critical Computing Education — Amy J. Ko

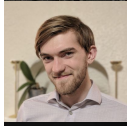
Greg Nelson he/him
greglnelson.info
HCI+CS Ed



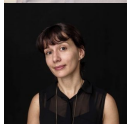
Benji Xie he/him
benjixie.com
HCI+CS Ed



Alannah Oleson they/them
alannaholeson.com
HCI+CS Ed

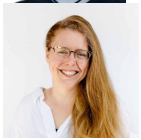


Neil Ryan they/them
neildryan.com
CS Ed



Stefania Druga she/her
cognimates.me
AI + CS Ed

Yim Register they/them
students.washington.edu/yreg
CS Ed + Data Science



Jayne Everson she/her
http://jayneeverson.com
HCI+CS Ed

The many wonderful doctoral students with which I collaborate.

Amy J. Ko, Alannah Oleson, Neil Ryan, Yim Register, Benjamin Xie, Mina Tari, Matt Davidson, Stefania Druga, Dastyni Loksa, Greg Nelson (2020). It's Time for More Critical CS Education. Communications of the ACM (CACM), 31-33. https://doi.org/10.1145/3424000