CS for What?

Diverse Visions of Computer Science Education in Practice

Rafi Santo, Sara Vogel
& Dixie Ching

CSforALL

ABOUT THE AUTHORS

Rafi Santo, PhD, is a learning scientist focused on the intersection of technology, education, equity and institutional change. He is a research associate at CSforALL and principal researcher at Telos Learning. Centering his work within research-practice partnerships, he has studied, collaborated with, and facilitated a range of organizational networks related to digital learning and computing education. His work at CSforALL involves supporting and researching school districts as they develop equitable computing education initiatives rooted in student and community needs, He's received support from the National Science Foundation, the Spencer Foundation, the MacArthur Foundation, the Mozilla Foundation, the Susan Crown Exchange and Google.

Sara Vogel is a doctoral candidate in urban education at the Graduate Center of the City University of New York, researching computer science education, bilingualism, and social justice pedagogy. She co-developed and is currently the lead research assistant on Participating in Literacies and Computer Science (PiLaCS), a National Science Foundation-funded project that leverages the diverse language practices of bilingual youth as resources in their computer science learning. A former bilingual teacher, she has also worked as a computing educator in informal settings with the nonprofit Global Kids Inc.

Dixie Ching, PhD, is a senior user experience researcher at Google Inc. Trained as a cell biologist and holding a master's degree in science journalism, Dixie has dedicated her career to helping people see the relevance of STEM in their lives and developing a better-informed public through STEM literacy. While a doctoral student in the Educational Communication and Technology program at NYU, she co-led Hive Research Lab, a research-practice partnership with the Mozilla Hive NYC Learning Network. Previously, she worked on research and design of educational media at the Joan Ganz Cooney Center at Sesame Workshop and the Center for Children & Technology/Education Development Center.

ABOUT CS FOR ALL

CSforALL is the national hub of the computer science for all movement with a mission to make high-quality computer science an integral part of K-12 education in the United States. Our three pillar approach; Support Local Change, Increase Rigor and Equity, and Grow the Movement, directs our work across a national and local spectrum to provide equitable and accessible K-12 computer science education to every student. We engage with diverse stakeholders leading computer science initiatives across the nation to support and facilitate implementation of rigorous, inclusive and sustainable computer science. For more information visit www.csforall.org or follow us on Twitter: @CSforALL.

ACKNOWLEDGEMENTS

We would like to thank the many educators from across New York City who shaped the foundational ideas in this white paper, too many to list individually here (see Appendix B). Special thanks goes to Leigh Ann DeLyser for her early and ongoing encouragement of this line of work. We appreciate many others who gave feedback on drafts of this report, including An-Me Chung, Hillary Kolos, Nathan Larson, Marc Lesser, Meghan McDermott, Amber Oliver, Tom O'Connell, Michael Preston, Julio Vasquez, and Jen Wilke. Thank you to LaToya Strong for opening up space for ideation at her March 2016 New York Coalition of Radical Educators Inquiry-to-Action Group, and to Global Kids and the CS4All team at the New York City Department of Education for use of their photos.

Support for this project was provided by the National Science Foundation award #1738675 and the Spencer Foundation. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or the Spencer Foundation.

Suggested Citation: Santo, R., Vogel, S., & Ching, D. (2019). CS for What? Diverse Visions of Computer Science Education in Practice. New York, NY: CSforALL.

Photo courtesy of New York City Department of Education Computer Science for All.

Educational plans and projects must have a philosophy... otherwise they are at the mercy of every intellectual breeze that happens to blow.

John Dewey, 1938

Contents

Values should drive what computer science education looks like.

Why should students learn computer science? For creativity? Jobs? Justice? Innovation? The answers to this question shape what computer science education (CSed) looks like in practice. CSed can and should look different, given the diverse aspirations and experiences of students, as well as their educators, families, and others who aim to support them.

And who is involved in articulating the visions driving CSed also matters because it has real implications for how the equity issues at the heart of the CS for All movement are addressed. Having stakeholders come together to do the messy and challenging work of hashing out their values makes it more likely that all needs and issues of equity can be addressed. It can create a "seat at the table" for broader voices to shape what CSed looks like on the ground.

This white paper provides a framework for understanding the myriad reasons for teaching computer science by considering their underlying values—beliefs about what is good, worthwhile, and important. It provides examples of how different CSed curricula, programs, and tools embody particular values and rationales in their design. Finally, our recommendations invite those involved in computer science education at every level—from policymakers and administrators to curriculum designers and teachers—to come together with students, families, and their communities to deliberate about why they think CS education is important, then use these answers to guide their choices around designing and making available CS learning opportunities.

When diverse stakeholders come together to consider, debate, and decide what values they want to drive their work around CS education, they embody the democratic ideals at the heart of education.

Photo courtesy of New York City Department of Education Computer Science for All.

Road Map

The CS Visions Framework

What values drive CS education?

The framework presented in the first section of this paper sheds light on the values behind common rationales for CS education, identifying seven core values that arguments for computer science education build on. Understanding these values can support the kind of community-based deliberation that's key to developing new initiatives around CS education.

CS Visions in Practice

How might values be translated into designs for CS education?

Considering the values behind CS education isn't just a philosophical exercise. Curricula and programs in CS are shaped by the values that drive them. The second section explores existing programs that embody diverse approaches to CS education, such as a focus on creative expression, broadening participation in computing, or using computing for social action. By describing each project's approach and unpacking the values they reference, these examples demonstrate how values get translated into educational experiences and designs.

Recommendations for a Values-Centered CS Education

How can different actors support bringing visions into practice?

Linking values to practice through an inclusive, deliberative process is more likely to yield projects that are aligned with the aspirations of communities, ones that empower and equip students to find success with computer science in the ways that stakeholders value most. To support communities in achieving these goals, the last section shares recommendations for CS education policy makers, instructional leaders, educators, content providers, and researchers.

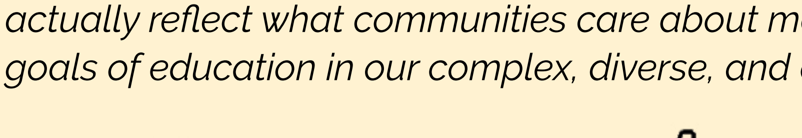
Photo courtesy of New York City Department of Education Computer Science for All.

A Note from the Authors

The fall of 2015 was a big moment for the CS for All movement in New York City. The city was the latest US locality to announce a new initiative to bring computing education to every child. Noticing this steady drumbeat of CS education policy talk, Sara Vogel, then a first-year grad student, shared frustrations with then-grad students Rafi Santo and Dixie Ching (now doctors) about the public discourse around these announcements. Journalists and politicians seemed to be framing the issue through one narrow lens: jobs, jobs, jobs.

As education professionals with backgrounds at the intersection of learning, equity, and digital culture in New York, Rafi and Dixie agreed there was an opportunity to bring the conversation in some new directions. All three of us knew so many dynamic educators, young people, and organizations that embodied a more diverse set of visions around what computing education could be for. We saw pedagogies that aligned with our values: using computing as a tool for creative expression, as a means of addressing injustices, as an avenue to joy, fun, and agency, or as itself a cultural space in need of transformation to better reflect voices that have been kept out. And importantly, we saw the need to create space to deliberate around how the values driving a major shift in the education system for tens of millions of students should be geared toward a vision of education as a public good—one serving the whole of society, for a broad range of impacts, rather than a private one serving more narrow interests.

We do this work because nothing is "values neutral" in education—every educational program, curricula, lesson, and learning tool has embedded within it certain purposes and valued ends for learners and for society. In understanding what those ends are, we can reveal the core values at play and who they are in service of. Being conscious of these values is a good in and of itself. It makes it possible for CS education efforts to actually reflect what communities care about most, bringing us closer to achieving the goals of education in our complex, diverse, and ever-evolving democracy.



Rafi, Sara, Dixie

Fall 2019

Photo courtesy of Hive Learning Network.

The CS Visions Framework

What values drive CS education?

The movement to bring computer science education to all students has quickly been embraced across the United States. Almost every state in the country is developing policies that aim to promote greater access to CS learning experiences, and at the local level, communities, districts and local education organizations have been involved in launching local initiatives focused on CSed.

On the ground, districts and schools embarking on CS for All initiatives have complex problems to solve: How will teachers get trained? What curricula should be used? How can we reach all students? What learning goals are appropriate at different ages? These are key questions. But a foundational question should not get lost in the rush to implementation: **Why should all students learn computer science?** Why should they, or their teachers, care? Why is CS education relevant to the issues people face in their communities?

Different answers to these questions will drive the development of different approaches to CS education. If stakeholders don't clarify the central purposes of CS for All initiatives, their efforts could wind up off course, losing sight of goals and priorities that really matter to their communities. To prevent that from happening, they need tools to support processes that clarify their values to answer the question: CS for what?

This section shares the **CS Visions Framework** (Vogel, Santo & Ching, 2017), a tool that outlines seven core values underlying the many rationales offered for computer science education. The framework can be used to identify the values underlying arguments for CSed and to spark conversation among members of a community about the role they see CS education playing in the lives and futures of young people, and in society more broadly.[1] The framework can also support design of new tools, curricula, and initiatives embodying these visions. When stakeholders know what their communities value, they can more thoughtfully make decisions about which CSed curricula, professional development models, and tools are the most aligned with those values.

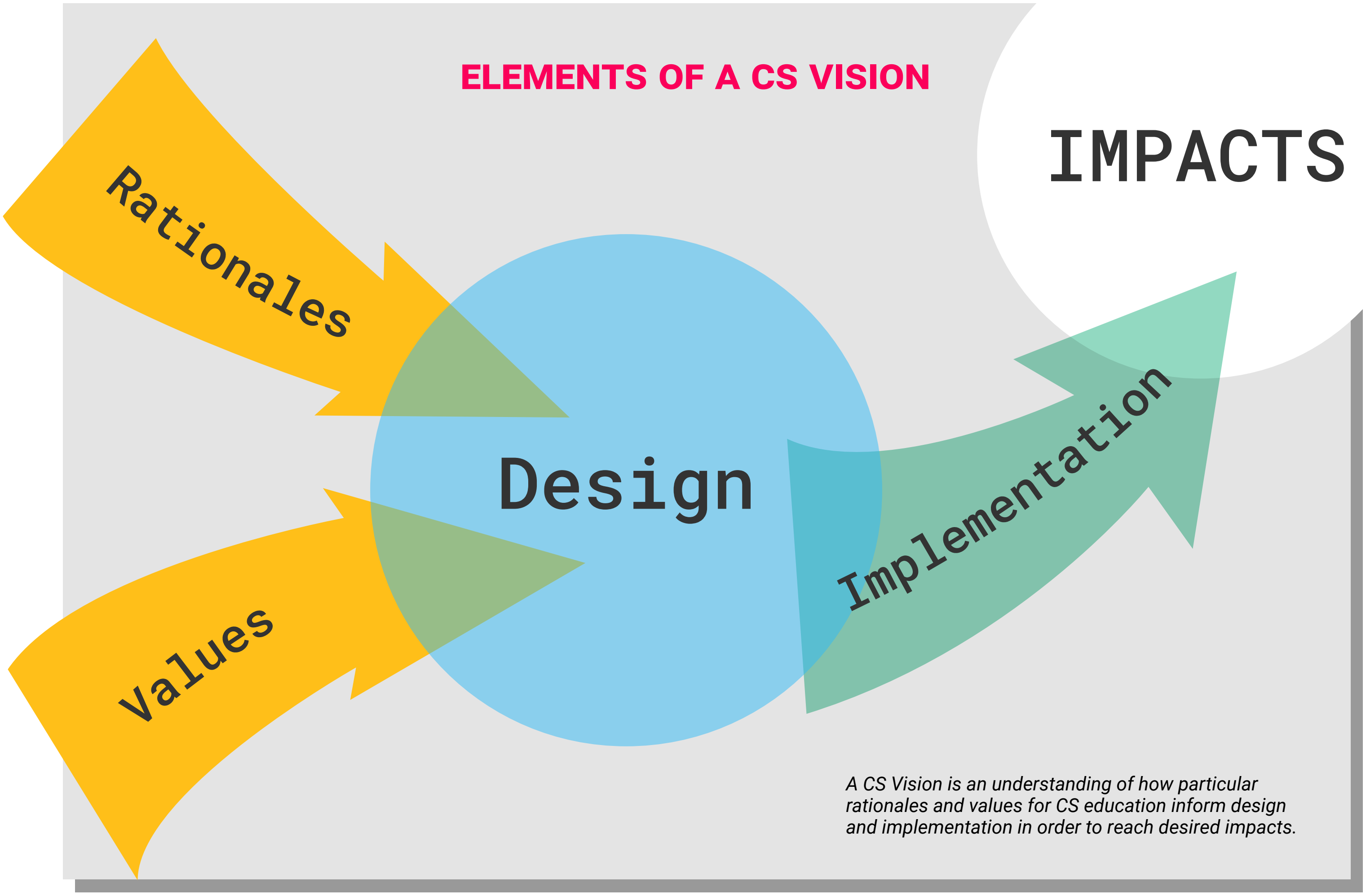
Why should all students learn computer science?

Different answers to this question will drive the development of different approaches to CS education.

Photo by Sara Vogel.

[1] Handouts and guides to help facilitate these kinds of conversations can be found at visions.csforall.org.

What is a "CS Vision"?



A CS Vision is an understanding of how particular rationales and values for CS education inform design and implementation in order to reach desired impacts.

Articulating a CS education vision means gaining clarity on five things. Identifying **rationales** for CS education—the why. Understanding that any given rationale has underlying **values** about what is important, and for whom. Rationales and values require aligned education **designs** and **implementation** strategies in order to reach intended **impacts** for individuals, community, and society—essentially, the changes that could result from CS education that's rooted in these values.

How did the framework come about?

The CS Visions Framework was produced through a collaborative knowledge-building process involving over 50 CS education stakeholders in New York City.[2] Together, stakeholders surfaced rationales for CS education—both common and less so—which the authors then synthesized into a coherent framework.

[2] See Appendix A for complete methodology and Appendix B for a list of contributors to the framework's development.

Seven core values and projected impacts of CS education

When people argue for CS education, they tend to draw on some combination of seven core values about the impacts it will or could have (below).

To be clear, these are not rationales in and of themselves, but are rather values and intended impacts that **underlie** people's rationales. For example, the statement that "CSed has the potential to usher in more project-based learning" reflects school reform and improvement as a value and impact area, while the idea that "we need CSed because youth shouldn't be just consumers, but producers of technology" expresses a value and intended impact around a certain type of citizenship.

Photo courtesy of New York City Department of Education Computer Science for All.

CS VISIONS – VALUES & IMPACT AREAS

Equity & Social Justice

Competencies & Literacies

Citizenship & Civic Engagement

Technological, Social & Scientific Innovation

Economic & Workforce Development

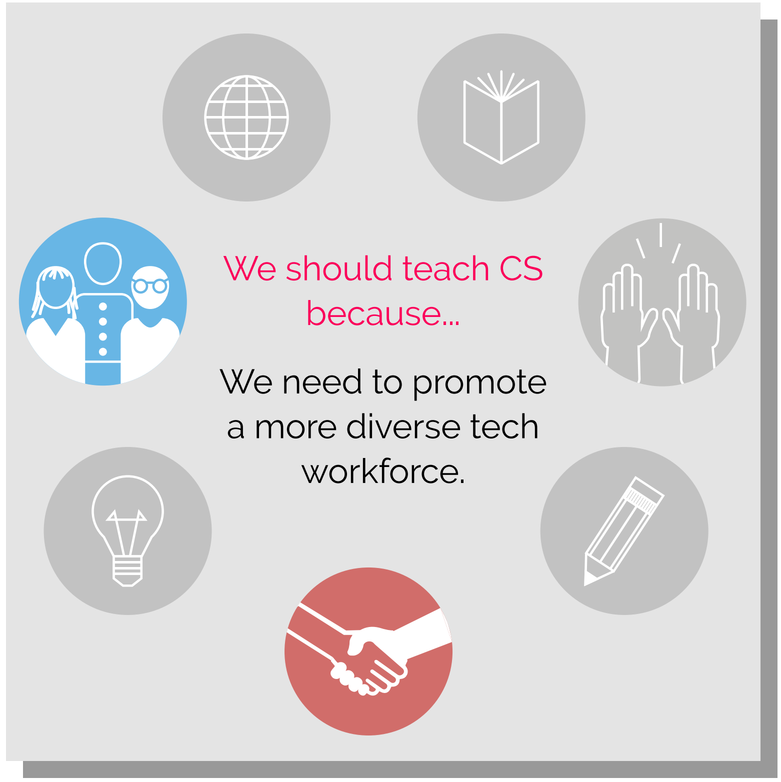
School Reform & Improvement

Personal Agency, Joy & Fulfillment

Using the framework to see values behind rationales

One of the most valuable things about the framework is how it acts as a sort of “x-ray goggles” to wear during conversations about CS education. It helps decode the ubiquitous, but also distinct, values that are present in people’s arguments.

For example, a rationale like “we need to diversify the tech ‘pipeline’ to make the tech sector more inclusive of people of color and women” projects impacts for industry as well as promoting ideals of equity and inclusivity. A comment suggesting that CS education would “teach youth to solve problems in their communities through technology” indexes values and impacts related to participation in civic life, production of technological innovations, and potentially making their world a more just place through those actions. The seven values and impact areas often sit alongside each other to varying degrees in a given rationale offered for computer science education.



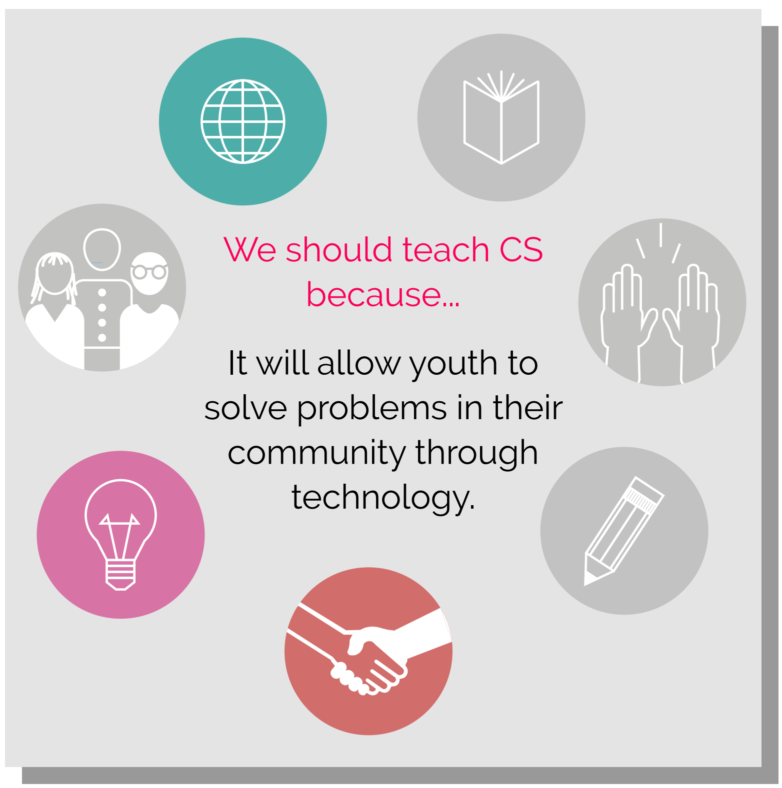


Photo courtesy of New York City Department of Education Computer Science for All.

All rationales are not created equal

On the pages that follow, we'll share sample rationales around each value and impact area. As you read, keep in mind that just because a reason is offered for computer science education does not mean that the assumptions behind it are sound. For instance, a common rationale associated with the competencies and literacies value is that teaching programming will improve students' general problem-solving skills. Research has found that this only happens if explicit scaffolds are in place to support learning transfer into other contexts (see Pea & Kurland, 1984, for this debate). There is also much controversy as to whether in fact the US is in fact experiencing a shortage of engineers that CSed might fill (Xue & Larson, 2015). For more information on CSed argument validity, see Burke, 2016 and Lewis, 2017.

Equity and Social Justice

Values and impacts that promote fairness and address historic injustices

SAMPLE RATIONALES

There are major disparities in minorities' and young women's engagement in STEM fields and universal CSed is part of addressing that.

It will level the playing field and help close the "digital divide" around tech for lower-income youth.

Not all tech will be in the best interest of our students—they'll need to be able to think critically about technology platforms. It's a "program or be programmed" world out there!

Our technology is largely designed by economically, racially, and socially privileged groups, and their biases and blind spots get embedded in our tech. CSed can help.

Competencies and Literacies

Values and impacts linked to computer science education's unique potential to promote particular skills

SAMPLE RATIONALES

It promotes 21st-century skills such as creativity, collaboration, and communication.

It has students engage in design thinking—identifying problems and then prototyping, testing, and iterating on solutions.

Knowing how to code is a new form of literacy.

It can promote systems thinking—the ability to understand and intervene in complex systems that are ubiquitous in our world.

Citizenship and Civic Engagement

Values and impacts focusing on computer science education's relationship to social, cultural, and political participation

SAMPLE RATIONALES

Being a good citizen in the 21st century will include digital citizenship.

Youth shouldn't just be consumers but also producers of technology.

Informed citizens need to understand the basics of how the technological world works in order to contribute productively to society as a whole.

Political and cultural participation are increasingly shaped by computing and our students need to understand social impacts of tech.

Technological, scientific, and social innovation

Values and impacts promoting scientific, technological, and social advancement, whether around current goals or ones currently not on the radar of those driving innovation

SAMPLE RATIONALES

It's important that people in local communities be equipped to address their own problems through having technologically fluent community members.

The more people we have who understand computer science, the more innovations and new knowledge we can produce as a society.

We need to produce scientific and technological innovations that solve "wicked" problems such as climate change and cybersecurity.

Technological innovation helps promote human flourishing—the next generation needs to know how to do it!

Economic and Workforce Development

Values and impacts that orient toward professional life, industry, and the economy

SAMPLE RATIONALES

There is a shortage of engineers and programmers that needs to be filled.

Computing may provide our youth with more and better career opportunities to choose from.

Computational thinking will be key no matter what career youth end up in.

It will strengthen local economies by attracting companies looking for technologically competent workers.

School improvement and reform

Values and impacts that see potential for CS education to work in service of broader school reform goals, benefitting teachers, school administrators, students, and the education systems they're a part of

SAMPLE RATIONALES

Teaching CS is a compelling new area that teachers are interested in and is a place where they can experiment with pedagogy.

CSed often uses project-based approaches that can enhance school pedagogy and move away from sage-on-the-stage approaches.

Practices from CS might enhance student learning of traditional academic disciplines (e.g., introducing computer simulation of an ecosystem to learn concepts in ecology, or using CS concepts to learn algebra).

Making technology is fun, so bringing CS to schools can increase student engagement.

Personal Agency, Joy and Fulfillment

Values and impacts oriented toward the fun, fulfillment, and personal agency that can come out of CS education as positive for their own sake

SAMPLE RATIONALES

Computing provides youth with the ability to express themselves creatively and have voice.

Being able to understand and make technologies gives kids power and agency.

Creating new technologies like apps, websites, or robots is fun!

The process of tinkering and making can lead to wonder, discovery, and enjoyment.

Using the CS Visions framework to guide deliberation

Beyond using the CS Visions framework to understand existing arguments and their underlying values, it can also be used to spark collective deliberation. Whether you are having conversations among district leadership teams, organizations that design CSed curricula or professional development, out-of-school organizations, funders, or policymakers, anyone involved in CS education efforts can do their work better when they have clarity about the visions they have for CS education. These conversations should not just happen at the beginning of an initiative, but throughout, as more is learned and achieved.

Going through existing arguments, like those offered on pages 14–17, and then debating, discussing, and surfacing those most important to a group can help establish a "north star" that can guide its work. Visit visions.csforall.org to find more unplugged activities and resources to support deliberation around values and rationales in CS education.

Using the CS Visions framework to guide design of CS education efforts

Perhaps the most important use of the framework is to ensure that CS education efforts are aligned with core values and toward intended impacts. "Efforts" here can refer to new CS lessons, curricula, after-school programs, or learning tools, but also broader CS education initiatives within schools and districts. The framework can help anyone—whether they have deep knowledge about CS education or not—make decisions about what type of CS they want to implement. Even someone without technical expertise can use their community's particular values as basis for evaluating CSed curricula and programs and choosing efforts that align with what they care about.

What does it look like to "align" the values and design of CSed efforts? The next section explores this more fully, but consider a school that says they want to teach CS so that students can express themselves creatively, but then largely uses coding puzzles with predetermined answers in their curriculum. Or, perhaps, the school has purchased materials for a creative makerspace but has not provided professional development for teachers on how to use the technology. We could also imagine a school that says it wants to teach CS to help close the gender divide in STEM fields, but then only offers CS as an elective and doesn't make any targeted efforts to recruit young women. Those struggles may be in part due to misalignment of vision and implementation.

If not designed intentionally, it's easy for CS education efforts to espouse rhetoric that doesn't actually inform implementation to achieve the impacts they seek.

Photo courtesy of New York City Department of Education Computer Science for All.

CS Visions in Practice:

Cases from the Field

How are values translated into design in CS education?

Arguments for computer science education and the values that undergird them have real implications for instruction, design, and implementation. Values—whether they are stated or implicit, well-thought-out or ad hoc—are baked into teacher professional development, curriculum, pedagogy, and learning technologies. While each CS tool, program, or curricula is unique, many have similar approaches and orientations. Some emphasize creative expression, others focus on equipping students for careers, some focus on computing's role in social action, etc.

This section showcases and describes various programs that share some common approaches when it comes to pedagogy, highlights examples within each, and unpacks the values they reflect.[3]

These examples show how visions—whether explicitly articulated or not—shape core components of CS education programs, impacting everything from who is taught, what tools are used, and what CS concepts are centered, to what partnerships are required, how learning is assessed, and more.

Some approaches to CS pedagogy

Creative Computing

Broadening Participation and Labor Empowerment

Computing Across the Disciplines

Computing for Social Action

EthicalCS and Learning Social Impacts of Computing

Photo courtesy of New York City Department of Education Computer Science for All.

[3] To be clear, we, the authors of this paper, identify values that these programs index based on our own analyses of program statements and materials in order to draw useful comparisons for readers. These and other CSed initiatives may be driven by values beyond what we describe, and program designers, educators, and the students that experience them may have other opinions about their underlying values. We encourage readers to take a look at our methodology in Appendix A and to do their own investigation of these and other CS initiatives.

Creative Computing

Some approaches focus on the expressive potential of CS education for students. Much like drawing, writing, or filmmaking, learning the language of code can be a powerful way for young people to express who they are, use their imaginations, and share their perspectives on their communities and the world. Looking at programs with this focus, it’s possible to see how how they bring these goals to life through the kinds of activities they engage their students in.

Photo courtesy of Eyebeam.

Imagining a more just world through code

EYEBEAM

Students can channel their creativity through CS tools to share their unique perspectives on issues they care about.

Eyebeam is a Brooklyn-based arts studio driven by three values: openness, invention, and justice. In their flagship youth program, Playable Fashion, teens design custom video game controllers and connect those controllers to a computer game of their design, coded using Unity 3D. When designing their game, youth are encouraged to answer questions like, "What would you fight for?" They create imaginative game controllers that they program to communicate with their game through physical computing components such as Arduino microcontrollers, motion and light sensors, and switches. Showcasing how he applied his creativity and imagination to express his voice, one youth participant created a side-scroller game in which the player is a skateboarder who turns police officers into hippies. The game controller resembled a skateboard with buttons corresponding to popular skateboarding tricks.

WE SHOULD TEACH CS BECAUSE...

Computing provides youth with the ability to express themselves creatively and have voice.

The process of tinkering and making can lead to wonder, discovery and enjoyment.

Tinker, Create, Share

Some CSed efforts value creative digital making for expression, personal enjoyment, and social learning.

Scratch is a creative coding environment where youth design interactive animations, games, and simulations and share their creations with others. Thousands of young people share new Scratch projects every day. In contrast with other platforms that offer more structured approaches (e.g., coding puzzles), the Scratch environment was designed to be a blank canvas to encourage students to see the "computer as paintbrush" (Resnick, 2006). Natalie Rusk's interviews with young Scratch users found they used the platform for five reasons: "to create, connect, share, learn, and have fun" (Rusk, 2016, p. 107). The design of Scratch, both as a tool and a community, emphasizes those values around personal agency and enjoyment through social learning. Any project can be remixed with the click of a button and users can comment on each other's projects. Youth can tinker and experiment with the code blocks and other components until their creation meets their goals, and Scratch supports "a diversity of project genres, so that people with widely varying interests can all work on projects they care about" (Resnick, 2014, p. 2).

WE SHOULD TEACH CS BECAUSE...

- Computing provides youth with the ability to express themselves creatively and have voice.
- Creating new technologies like apps, websites or robots is fun!
- Creative and blocks-based approaches can democratize access to learning CS concepts and practices and lower barriers to entry.

Exploring identity with CS tools

PARTICIPATING IN LITERACIES AND CS (PILACS)

Students can use CS tools to tell their stories and seek answers to questions like, “What has made me who I am?” and “What communities am I part of?”

As part of the research practice partnership Participating in Literacies and Computer Science, a bilingual sixth-grade teacher in New York City and university-based researchers developed a CS-integrated social studies project called “Journeys to School” (Ascenzi-Moreno, Guílamo & Vogel, forthcoming). To kick off the project, students reflected on a series of images depicting the journeys to school of children around the world and the obstacles that students often face. They then created games using the Scratch platform to depict their own journeys to school—in New York, but also in the countries where many of them had recently immigrated from. Instruction was delivered in the languages of the students (in this case, Spanish and English) to help students truly see themselves in the material. The project also offered students an opportunity to reflect on their own experiences, obstacles they’ve faced, and identities as they prototyped and built their games. They incorporated images and elements from their neighborhoods and schools as they learned about CS concepts such as events and behaviors and engaged in CS practices such as debugging and sequencing.

WE SHOULD TEACH CS BECAUSE...

If taught in culturally and linguistically responsive ways, young people can come to see computing as a vehicle for telling their stories.

Computing provides youth with the ability to explore their identities.

Broadening Participation and Labor Empowerment

The most common rationales around expanding CSed focus on equity in who gets to participate in the computing sector. Broadening participation of underrepresented groups in CS has been motivated by both an equity imperative to provide access, as well as concerns that a lack of diversity leads to a tech sector only focused on certain problems. Others are more focused on careers in technology as a potential source of economic mobility, regardless of anything else. Each of these rationales reflects related values, but, as we explore here, slight differences in values lead to programs that look quite different in practice.

Photo courtesy of Black Girls Code.

Diversifying Computing Fields and Practices

Some programs value providing access for underrepresented groups to computing practices, workplaces, and fields. To accomplish this, they pay careful attention to who is participating and how to engage them in learning.

Programs supporting specific underrepresented groups, such as the youth-serving organization Black Girls Code, are rooted in the value of broadening participation. Programs like BGC aim to increase underrepresented groups' participation in STEM fields and the computing industry. To achieve this, they follow shared principles such as providing mentors who work in the technology sector who come from similar backgrounds and colocate events and programs at the offices of technology companies. In another approach to broadening participation, the Exploring Computer Science curricular initiative focuses less on professional pathways. Their culturally responsive curriculum and teacher training program instead aims to expand access to cultural and social participation in computing, "democratiz[ing] computer science knowledge [...] with a specific focus on traditionally underrepresented groups" (ExploringCS.org, n.d.).

WE SHOULD TEACH CS BECAUSE...

Computing may provide our youth with more and better career opportunities to choose from.

There are major disparities in minorities' and young women's engagement in STEM fields and universal CSed is part of addressing that.

Labor Empowerment

THE KNOWLEDGE HOUSE

For some, CSed is more about economic opportunity than broadening participation in CS, with programs intentionally linking economically disadvantaged youth to job opportunities in tech.

The Knowledge House, located in the South Bronx, runs free bootcamp-style tech programs focused on job access for low-income youth. They offer advanced courses in areas such as full-stack web development, data science, and user experience/user interface design, and actively work with local community colleges and employers to actively coordinate entry into professional pathways. They choose to focus on the tech sector since there are more job opportunities that don't require bachelor's degrees, a reality for many of their youth, and talk about participation in computer science as a means to economic self-determination. The ultimate goal for TKH is "bootstrapping a technology ecosystem in the South Bronx", as co-founder Joe Carrano puts it, that can provide a stable, self-reliant economic base for a community that has historically been a site of disinvestment.

WE SHOULD TEACH CS BECAUSE...

Computing may provide our youth with more and better career opportunities to choose from.

Computing can provide a stable, self-reliant economic base to communities that have been historically marginalized.

Transforming Industries through Diversity

The goal around broadening participation doesn't necessarily stop at providing access to computing industries, but can focus on changing what kinds of problems these industries focus on.

A game called Tampon Run (Tiku, 2015), created by New York City girls participating in Girls Who Code, focused on breaking taboos around menstruation. The game was eventually fully developed for iOS, with a bit of help from a development company that Girls Who Code staff had relationships with. The girls that created it shared that they didn't think they would have been willing to run with the idea if they'd been in a co-ed program. The example shows that broadening participation programs can not only be about changing who is involved in the technology sector, but what kinds of problems, and associated social and technological innovations, might be possible to address within it.

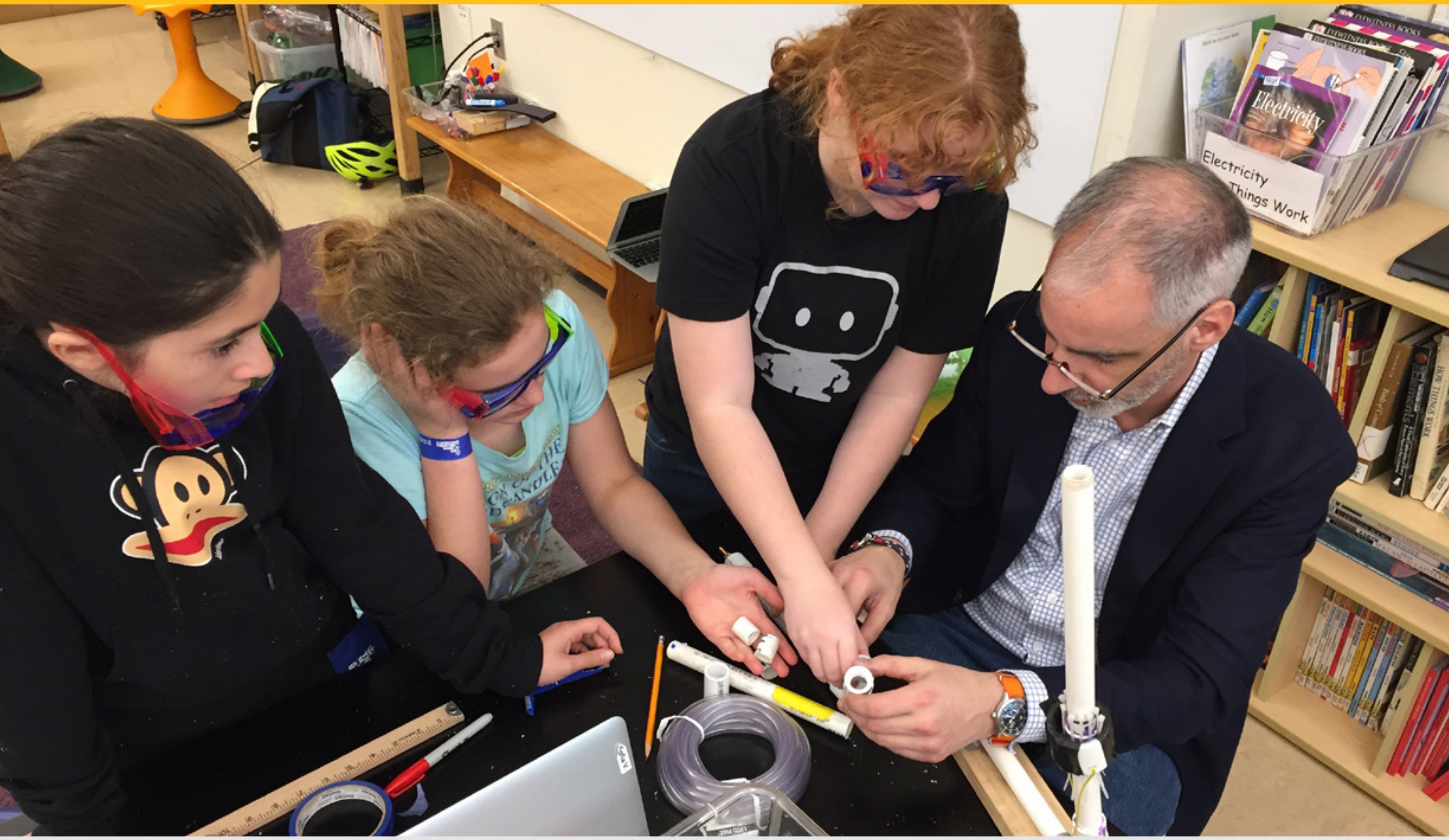
WE SHOULD TEACH CS BECAUSE...

We can expand the range of problems solved through tech to encompass issues faced by those marginalized in CS.

Computing may provide our youth with more and better career opportunities to choose from.

There are major disparities in minorities' and young women's engagement in STEM fields and universal CSed is part of addressing that.

Computing Across the Disciplines



Districts across the nation are working to integrate CS education into the formal school day. Since computing education incorporates skills and competencies around 21st-century technologies, computational thinking, collaboration, and critical thinking, some programs use it to enhance student learning in the traditional academic disciplines such as math, science, and language arts. This sub-section describes initiatives which support schools to integrate computer science concepts and practices into other subjects—grounded in the vision that CS is useful to broad audiences of students as a tool for inquiry in other fields.

Photo by Karen Blumberg.

Enhancing Math Learning, Broadening Participation

By integrating into and enhancing traditional school subjects, some programs aim to reach more diverse groups of students.

The Bootstrap math curriculum is designed to bring algebra alive for students in ways that static textbooks can't: through game design (Felleisen & Krishnamurthi, 2009). Bootstrap's founders have described algebra as a "gatekeeper" for postsecondary education, and to address this, explicitly align the programming concepts involved in game design to algebra to minimize confusion over, for instance, what can be differing definitions of a "function" in math and the definition of a "function" in CS (Schanzer, Fisler, Krishnamurthi, & Felleisen, 2015). Indexing its commitment to student learning in math, the organization has mostly supported math teachers without a CS background, building their capacity through professional development workshops. Bootstrap also measures its success by tracking student scores on algebra assessments (Bootstrapworld.org, n.d.). The program's founders argue that to truly reach all students—including those typically underrepresented in STEM and computing—CS education must meet them in their mandatory academic courses and not in electives (Schanzer, panel presentation, 2017). Bootstrap's participation statistics bear this out: according to Bootstrap's teachers, 43% of students enrolled in their courses are female and 46% are members of underrepresented minority racial groups (Bootstrapworld.org, n.d).

WE SHOULD TEACH CS BECAUSE...

If included in mandatory academic courses, CS education will reach those typically underrepresented in computing.

Practices from CS might enhance student learning of traditional academic disciplines.

Inquiry-Driven Learning in Science Through CS

PROJECT GUTS
Growing Up Thinking Scientifically

Some curricula integrate CS to bring traditional science teaching and learning up-to-date, recognizing that in the 21st century, computing offers key tools for the kinds of inquiry and experimentation that form the heart of science.

Project GUTS is a curriculum that helps students use, modify, and create scientific models using a web-based software called StarLogo Nova, developed at MIT. In one learning module, students do a deep dive into how an ecosystem's energy flow works by programming simple rules for "agents" in a model to follow—rabbits eat grass, lions eat rabbits—running experiments using the model, and collecting data about how the system changes when the conditions and rules change—what if the rabbits move faster, or the lions have to rest between kills?

Aligned to the Next Generation Science Standards, Project GUTS aims to engage students in doing science, not just memorizing facts. Its creators believe that today, CS concepts and practices are key to conducting science across many disciplines (Lee, panel discussion, 2017). Along with valuing computing as a means to support science learning generally, Project GUTS values particular competencies and literacies, including systems thinking (Weintrop et al., 2016). They also value supporting youth to engage in technological, social, and scientific innovation—as they put it, how "to look at the world and ask questions, develop answers to the questions through scientific inquiry, and design solutions to their problems" (projectguts.org, n.d.).

WE SHOULD TEACH CS BECAUSE...

It can promote systems thinking - the ability to understand and intervene in complex systems that are ubiquitous in our world.

Practices from CS might enhance student learning of traditional academic disciplines.

Computing for social action

Another approach to CS education guides youth to use computing to address social issues—local, global, and otherwise. Despite sharing some similarities, these programs are distinct in their implementation partly because they also index a host of other values for CS education—for instance, broadening participation, developing youth agency, creating technical social innovations, or civic engagement.

Courtesy of Global Kids, photo by Sara Vogel.

Promoting civic participation through tech and open data

When open data sets about communities are integrated into computing ed, programs can promote civic engagement and participation.

The New York City Department of Education, in collaboration with the nonprofit Beta NYC, runs a series of Hack League hackathons, which invite middle and high school students to apply CS concepts and practices to open data sets from New York City digital archives. Students use these data sets to create CS projects around issues like quality of life, the environment, and housing in their neighborhoods.

WE SHOULD TEACH CS BECAUSE...

Students can mobilize computing tools and data to better participate in political and civic life.

Using data to participate in civic life can leave youth with a sense of empowerment.

Entrepreneurship with and for communities

CSed can expand the range of problems solved through tech to encompass the everyday issues faced by participants not traditionally represented in CS and engineering.

The Technovation Girls promotes tech entrepreneurship and innovation for young women and girls around the world with an annual contest to create mobile apps to solve real community problems. Youth receive mentorship from female role models and are expected to write a business plan and present a pitch to accompany their app submissions. The organization tracks its impact not just by how many young women become interested in CS and STEM education and careers, but by the kinds of apps created by participants, which in 2013-15 addressed issues in education and learning, health and fitness, community connection, counseling, and volunteerism. An app created by young women in Mombasa, Kenya, for instance, was designed to facilitate the recycling of materials people would ordinarily throw away, given that region's challenges in sanitation (Technovation, 2016).

WE SHOULD TEACH CS BECAUSE...

We can expand the range of problems solved through tech to encompass issues faced by those marginalized in CS.

It is a context for teaching skills needed for the workforce, like writing pitches and business plans.

It offers girls traditionally underrepresented in STEM an opportunity to participate.

Youth Agency and Empowerment

Programs housed within youth development organizations engage youth in creating socially focused technologies to foster agency and empowerment, and skills such as collaboration and communication.

Courses designed by the national nonprofit Mouse engage youth in user-centered design to produce green technologies, or assistive devices for people with special needs. The Playing for Keeps program at the NYC-based youth development organization Global Kids asks students to design and code games about local and global social issues to raise awareness. While these organizations hold values around developing technological innovations that might be impactful, the programs are more substantively guided by a philosophy and pedagogy that sees youth design of technologies addressing local, national, or global issues as an avenue for them to become leaders and agents of social change.

WE SHOULD TEACH CS BECAUSE...

It can foster youth agency and empowerment.

Youth can design new technologies centered around the needs of particular users or to solve "wicked" challenges.

When youth design and build technologies that address social issues, they become agents of change.

Social action projects as gateways to CS fields

Leveraging issues youth care can be an entry point to CS for students who might not have considered taking part. In this case, computing for social impact is also a means of achieving goals around broadening participation.

One curriculum aligned to the Advanced Placement test, Mobile CS Principles, expects high-schoolers to design and program "socially useful" mobile apps, write and talk about their ideas, and collaborate with peers. This curriculum leverages students' interests in social issues to demonstrate that there might be a place for them in CS fields.

WE SHOULD TEACH CS BECAUSE...

CSed offers a unique context for learning how to collaborate and communicate.

If taught in ways that leverage social issues, CS education can broaden participation.

Programming can be a tool for expression.

EthicalCS and Learning Social Impacts of Computing

Computing has become an integral part of society, from commerce and communication to politics and entertainment. The consequences of this have not always been positive—loss of privacy, disinformation that fractures public discourse, and biased algorithms reinforcing existing inequalities have led to calls to educate youth on these issues within the context of K12 CS education. Rationales here include the need for a technologically sophisticated citizenry that can respond to social changes and the ability for students to "push back" against potential negative effects of tech. This section highlights programs that center these values around ethicalCS and having students understand the social impacts of computing in the design of CSed efforts.

Photo courtesy of New York City Department of Education Computer Science for All.

Understanding Privacy and Security

Having students learn about privacy and security—maintaining understanding of and power over our personal data in terms of who has access to it and what it's being used for—offers opportunities for students to engage in activities that focus on their own online lives.

The Mozilla Foundation, creator of the Firefox web browser and advocate for web literacy, has created curricula addressing issues such as password security, understanding how third parties track our online behavior, protecting personal privacy online, and assessing risk when it comes to personal information and data.

In one activity, students create "data trail timelines"—brainstorming all the different moments and ways they participate online throughout a day—then reflect on what sort of data might be collected about their online behavior and by whom. In another, students use a tool called Lightbeam that visualizes third parties that track data from sites they visit, having them reflect not just on issues of privacy, but its relationship to surveillance-based business models associated with the online world.

WE SHOULD TEACH CS BECAUSE...

Every young person is growing up in a world where their privacy is affected by tech. They need to understand how to protect their privacy and security online.

The business models of many tech companies rely on surveillance of our personal lives, and kids need to understand the ways their data is used, and potentially, abused.

Critiquing Technology as Literacy

For some, the ability to critique the designs of technology is a new form of literacy—both to support advocacy around existing platforms, as well as to support a generation of new designers who think critically about the impacts of what they're making.

COMPUGIRLS, a culturally responsive technology program for adolescent girls from under-resourced schools, focused on developing skills around what it calls "techno-social activism," "a form of activism that trains girls of color to recognize the affordances and limitations of technology and to have a critical perspective on how technology can be used for social change" (Scott & Garcia, 2017, p. 67). In one activity, girls create a 3D avatar in the Second Life platform to jump-start discussions about how technology platforms may support, or not, representation of those of different racial backgrounds and identities. The goal is for girls to see how the design of a digital tool can be limited by the worldview of the tool's creators.

Similarly, in a precollege course titled "Tech for Good: Designing Ethical Innovations," New York Times technology and education reporter Natasha Singer asks students to interrogate possible harmful outcomes associated with design choices they make in the context of developing "social good" oriented apps.

WE SHOULD TEACH CS BECAUSE...

Youth can come to advance a vision in society where the success of a technology is judged on "who creates, for whom, and to what ends."

If taught in culturally responsive ways, young women of color can develop critical perspectives and use technology for social change.

Promoting Ethics in “Big Data”

With the rise of “big data,” many see it as critical that students understand the ways that bias can be introduced into data sets, as well how data gathering intersects with issues of ethics.

In BootStrap’s Data Science curriculum, students learn about ethical reporting of data and ways that data can be abused in ways that harm. They learn about ways that bias can play a role in the sampling and analysis of data and, in the process, reinforce social inequalities.

The program’s creators share that, “By becoming data-critical consumers and producers, and understanding the role that data plays in the world around them, we help them develop the self-confidence and skills to ask questions, probe deeper when data is presented, and be constantly aware of what agenda may lurk behind the numbers.”

WE SHOULD TEACH CS BECAUSE...

Regardless of whether they going into computing careers, all kids will need to be critical consumers and producers of data in the 21st century.

“Big data” can have biases that lurk behind the numbers, which can result in reinforcing inequalities when we design technologies based on these skewed data. Students, especially those going into tech, need to understand data ethics.

Promoting EthicalCS through Learning Social Impacts of Computing

There are limited efforts to develop K12 curricula that support learning around the social impacts of computing, but broader trends in this area point to areas CSed community can draw on to help develop a more ethical and accountable tech sector.

In higher education, there's increasing attention being given by college and university computer science programs to ethics in computing, and an open-access repository is maintained by CU Boulder faculty Casey Fiesler around tech ethics courses at the university level.

While there not as many resources in this area in K12, there are emerging efforts to address this, such as the #ethicalCS group and monthly Twitter chat that has served as a space for discussion around how design of networks, algorithms, and processes of abstraction might implicate ethical issues and how to engage in classroom conversation around these topics.

In broader work outside of K12 education, we see efforts such as the Algorithmic Justice League, which highlights instances of algorithmic bias; the Center for Humane Technology, which focuses on the negative effects of monetizing attention through Internet platforms; and a range of books that explore these issues, such as Cathy O'Neil's Weapons of Math Destruction (2016), James Bridle's New Dark Age (2019), Virginia Eubanks' Automating Inequality (2018), and Safiya Noble's Algorithms of Oppression (2018).

Going forward, the K12 CSed community has the opportunity to build on the increased attention to this issue that's reaching the public discourse.

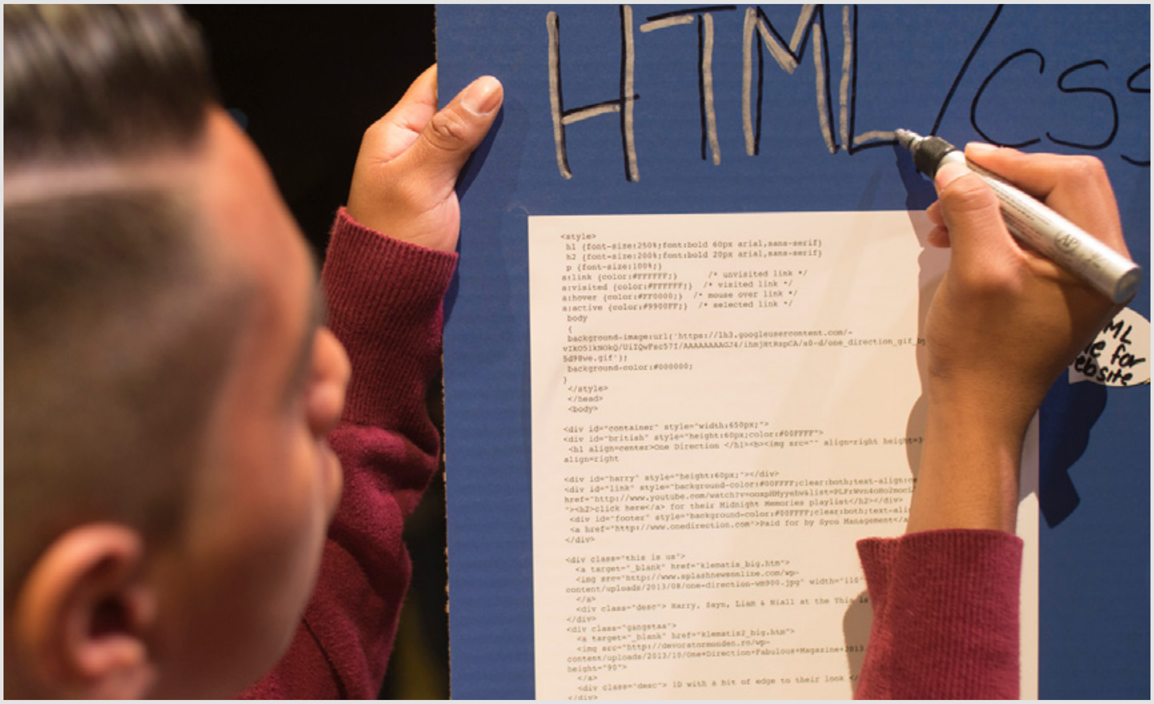


Photo courtesy of New York City Department of Education Computer Science for All.

Photo courtesy of New York City Department of Education Computer Science for All.

Recommendations for Values-Centered CSed

How can different actors support bringing visions into practice?

As the previous section showed, different values result in quite different practices on the ground when it comes to teaching and learning CS.

A vision rooted in labor empowerment may require youth-serving organizations or schools to partner with local businesses. One rooted in broadening participation may entail making connections to racial or gender affinity groups within science and technology sectors. A vision rooted in technology for social impact may require identifying real-life community issues and problems for students to address through their designs. A vision rooted in creative computing may require hiring teaching artists to collaborate with students, or curricula that involve more open-ended projects.

> There is not one purpose for education, and there should not be one purpose for computer science education either.

Indeed, even if a given CS education program or curriculum centers on one particular purpose, we'd hope that across the ecosystem of learning opportunities a young person experiences a plurality of purposes emerges, ones keyed to the many ways we think about education's role, from fostering creativity and agency, preparing youth for a changing professional landscape, opening up opportunity where it's been closed off, and supporting the emergence of critical thinkers and civic-minded community members.

Given that holding different visions has important implications for design and implementation of CSed efforts, we encourage CSed stakeholders at all levels to actively reflect on and collectively deliberate around what values drive their efforts and how designs can best index those values.

In this section, we list some specific recommendations for CS education stakeholders at many levels, including policy makers, instructional leaders, educators, content providers, and researchers.

Recommendations for Policy Makers

Policy makers have key roles to play in supporting vision-setting for CS education. In general, the role of the policy maker around CSed visions is to ensure policy communicates and supports diverse visions of CS education in an explicit and aligned way.

1 Articulate the CS visions undergirding your state or district's adopted standards to highlight how standards can help advance local visions.

2 Amplify conversations about "why CS" through your communications efforts. Talk about a wide range of impacts, and not just those that may be the most politically expedient.

3 When standards come up for revision, convene diverse stakeholders and consider how the standards are, or are not, supporting local visions of CS education.

4 Ensure that standards documents leave enough room for districts, communities, and educators to self-determine their own visions for CS education, so that the resulting learning goals, pedagogies and programs reflect local needs and priorities. Create structures that support local education agencies in your locality to understand and articulate CS visions tied to their community needs, and highlight examples of different visions that emerge.

5 Ensure that the visions of local communities are not only well resourced, but are resourced in a way that's aligned to the intended impacts. This can mean ensuring that curriculum and professional development are linked to articulated purposes for CS, and are consistent with one another.

Recommendations for Educators and Instructional Leaders

The core role of instructional leaders and educators is addressing the question of "Why CS education?" is to learn about the CS education visions of your school community, and then align your efforts to make those visions a reality.

1. Learn more about your students', families,' community partners', and colleagues' visions and interests, asking them why they think it's important to learn CS. Find resources to support these conversations at visions.csforall.org. Aim to bring as many voices "to the table" as possible, and to actively expand the range of purposes that CS education may serve, given that different students will connect with different kinds of CSed.

2. Use the visions of your community to help vet, align or create new curricula, professional development opportunities, assessments, tools, afterschool programs, and partnerships.

3. Resist the temptation to purchase flashy tools that don't align with your visions and won't meet your community's needs.

4. Align expected outcomes, frameworks for coaching teachers and assessing student work to the school community's visions for CS education.

5. Ensure the school community's values around CSed are reflected in the classroom level experiences of students.

Recommendations for Content Providers and Tool Designers

The framework in this document aims to inspire content providers, such as curriculum developers, technology designers and professional development providers, to reflect on the visions and values embedded in their designs.

1 Reflect on how your offerings index different values to gain clarity about your CS visions internally.

2 Make your CS visions transparent, so that educators and leaders can make more informed decisions about the tools and curricula that would work best for them.

3 Consider how well your current offerings deliver on their intended impacts and align with the values that you state around CS education.

4 Consider which visions are not as well-represented in the current landscape and work with communities to fill those gaps with new designs.

5 Support educators to understand your vision so they might thoughtfully and intentionally implement your tool or curricula for their context.

Recommendations for Researchers

There are key questions to study regarding the implementation of CS for All efforts. Just as important are questions related to the "why" behind implementation which this white paper explores.

1. Conduct research that sheds light on the assumptions behind various arguments for CS education, and whether and under what conditions these rationales have validity.

2. Investigate ways that different CSed efforts can effectively instantiate values in ways that reach intended impacts, and what gets in the way of doing so.

3. Surface and potentially critique dominant values within CS education efforts. Attend to understudied and less dominant visions around the purposes of CS education.

4. Understand the dynamics of how CS visions form and shift at different levels within the CSed world (e.g. within organizations designing content and professional development, within schools and districts, across community groups and settings).

5. Use research to uncover principles that support effective deliberation among CSed stakeholders, with attention to how actors coming from different power positions can have authentic seats at the table.

The CS for All movement needs to move beyond simply pushing for “more CSed” and towards being in conversation with communities to determine *what kinds of CSed* will best help them realize their goals.

Conclusion

Photo courtesy of New York City Department of Education Computer Science for All.

Computer science for what?

Communities across the US are different, face different challenges, have different strengths and resources, and as such, may see CS education as a means to diverse ends. It was meaningful for students in the rural parts of the Mendon/Upton Regional School District in Massachusetts to use computer science concepts and practices to engineer and program a revenue-generating composting system for the horse manure produced on nearby farms (Kadi & Quinn, 2018), valuing both economic as well as environmental sustainability. Equally meaningful was a project completed by students at a software engineering-themed high school in the Bronx, New York, to create an app and a game to raise awareness about stop and frisk practices of police (Hu, 2016), valuing CS as a means to promote social justice through technological innovation. In short, different contexts hold different answers to the question "Computer science for what?"

To be clear: the examples above should not give the impression that the framework should be used to pigeonhole communities into choosing only certain types of CS learning opportunities. In fact, our framework demonstrates that CS education is a tool for engaging with the world beyond students' direct experiences, whether that is for labor empowerment, creative expression, social change, civic engagement, to deepen exploration in an academic subject, or to examine the impacts of technology in our world. Our objective in offering this framework is to promote inclusive, reflexive and values-based decision-making for CS education at all levels. We believe this kind of process is more likely to yield initiatives that hit the sweet spot: vision-driven, rigorous, and equitable CS learning experiences for all. This means deliberating in a way that supports community members from different backgrounds to contribute, and to see their goals represented within local initiatives. This way, stakeholders can better attend to blind-spots, ward off provincialism, and question implicit assumptions about what is possible, leading to models that are ultimately more inclusive and expansive.

Reflexivity and inclusion for CS education advocates, researchers, and content providers means moving beyond simply pushing for "more CSed" and towards being in conversation with communities to determine the particular forms and even "genres" of CSed that are the most desired, and what's needed to support those visions in practice.

References

Ascenzi-Moreno, L., Guilamo, A., & Vogel, S. (Forthcoming). Integrating Coding and Language Arts: A View into Sixth Graders' Multimodal and Multilingual Learning. *Voices from the Middle*.

Bilkstein, P. (2018). *Pre-College Computer Science Education: A Survey of the Field*. Retrieved from Google LLC website: http://services.google.com/fh/files/misc/pre-college-computer-science-education-report.pdf

Bootstrap: Impact. (n.d.). Retrieved March 15, 2019, from https://www.bootstrapworld.org/impact/

Brennan, K., Balch, C., & Chung, M. (2014). *Creative Computing: Scratch Curriculum Guide*. Retrieved from scratched.gse.harvard.edu/guide/

Bridle, J. (2019). *New Dark Age: Technology and the End of the Future (Reprint edition)*. London ; Brooklyn, NY: Verso.

Burke, Q. (2016). Mind the metaphor: Charting the rhetoric about introductory programming in K-12 schools. *On the Horizon; Bradford, 24*(3), 210–220.

Can Math Teachers Teach Computer Science? (2015, October 9). Retrieved March 15, 2019, from Worcester Polytechnic Institute website: https://www.wpi.edu/news/mathcsci

Dewey, J. (1938). *Experience and Education*. Kappa Delta Pi.

Eubanks, V. (2018). *Automating Inequality: How High-Tech Tools Profile, Police, and Punish the Poor*. New York, NY: St. Martin's Press.

Exploring Computer Science. (2010, August 31). Mission. Retrieved March 15, 2019, from Exploring Computer Science website: http://www.exploringcs.org/about/mission

ExploringCS.org. (N.D.). Exploring Computer Science Mission. Retrieved September 21, 2019, from http://www.exploringcs.org/about/mission

Hu, W. (2016, February 21). Teaching Bronx Students the Language of Computers. *The New York Times*. Retrieved from https://www.nytimes.com/2016/02/22/nyregion/teaching-bronx-students-the-language-of-computers.html

James, S. M., & Singer, S. R. (2016). From the NSF: The National Science Foundation's Investments in Broadening Participation in Science, Technology, Engineering, and Mathematics Education through Research and Capacity Building. *CBE Life Sciences Education*, 15(3). https://doi.org/10.1187/cbe.16-01-0059

Kadi, A., & Quinn, D. (2018). *That's Horse S#!T: Connected Learning Through Compost, Weather Balloons & Other Projects*. Presented at the Connected Learning Summit, Boston, Massachusetts. Retrieved from https://connectedlearningsummit2018.sched.com/event/Egz1/thats-horse-st-connected-learning-through-compost-weather-balloons-other-projects

Lee, I. (2017, November). *Computing Integration Panel*. Presented at the To Code and Beyond: Rigor & Joy, Cornell Tech, New York, New York.

Lee, R. (2013). Teaching Algebra through Functional Programming:An Analysis of the Bootstrap Curriculum. *All Theses and Dissertations*. Retrieved from https://scholarsarchive.byu.edu/etd/3519

Lewis, C. (2017). Good (and bad) reasons to teach all students computer science. In S. B. Fee, A. Holland-Minkley, & T. Lombardi (Eds.), *New Directions for Computing Education: Embedding Computing Across Disciplines*. New York: Springer.

McClanahan, W. S., Peppler, S. K., & Polin, M. (2016). *"I program my own videogames": An Evaluation of Boostrap*. Retrieved from http://www.bootstrapworld.org/research/EvalReportDec2015Medium.pdf

Next Generation Science Standards. (n.d.). Retrieved March 15, 2019, from https://www.nextgenscience.org/

Noble, S. (2018). *Algorithms of Oppression: How Search Engines Reinforce Racism* (1 edition). New York: NYU Press.

O'Neil, C. (2017). *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy* (Reprint edition). New York: Broadway Books.

Pea, R. D., & Kurland. (1984). On the cognitive effects of learning computer programming. *New Ideas in Psychology*, 2(2), 137–168.

Resnick, M. (2006). Computer as paint brush: Technology, play, and the creative society. *Play= Learning: How Play Motivates and Enhances Children's Cognitive and Social-Emotional Growth*, 192–208.

Resnick, M. (2014). Give P's a chance: Projects, Peers, Passion, Play. *Proceedings of Constructionism and Creativity Conference*. Presented at the Vienna, Austria. Retrieved from http://constructionism2014.ifs.tuwien.ac.at/papers/1.2_1-8527.pdf

Rusk, N. (2016). Motivation for Making. In K. Peppler, E. R. Halverson, & Y. Kafai (Eds.), *Makeology* (Vol. 2). Retrieved from https://www.taylorfrancis.com/books/9781317537090/chapters/10.4324/9781315726496-6

Santo, R., Ching, D., Peppler, K., & Hoadley, C. (2017). Participatory knowledge building within research-practice partnerships in education. In *SAGE Research Methods Cases*. London, United Kingdom: SAGE.

Schanzer, E. (2017, November). *Computing Integration Panel*. Presented at the To Code and Beyond: Rigor & Joy, Cornell Tech, New York, New York.

Schanzer, E., Fisler, K., Krishnamurthi, S., & Felleisen, M. (2015). Transferring skills at solving word problems from computing to algebra through bootstrap. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 616–621. ACM.

Scott, K. A., & Garcia, P. (2017). Techno-Social Change Agents: Fostering Activist Dispositions Among Girls of Color. *Meridians: Feminism, Race, Transnationalism, 15*(1), 65–85.

Technovation. (2016). *Technovation Three Year App Analysis*. Retrieved from https://technovationchallenge.org/wp-content/uploads/2016/08/TechnovationAnalysys-2015-1.pdf

Tiku, N. (2015, February 3). Teenage coders behind Tampon Run take their feminist game to the App Store—The Verge. *The Verge*. Retrieved from https://www.theverge.com/2015/2/3/7964667/tampon-run-feminist-game-girls-who-code

Vaidyanathan, S. (2017, November 16). Why Computer Science Belongs in Every Science Teacher's Classroom—EdSurge News. *EdSurge*. Retrieved from https://www.edsurge.com/news/2017-11-16-why-computer-science-belongs-in-every-science-teacher-s-classroom

Vogel, S., Santo, R., & Ching, D. (2017). Visions of Computer Science Education: Unpacking Arguments for and Projected Impacts of CS4All Initiatives. Proceedings of the 47th ACM Technical Symposium on Computing Science Education. Presented at the Seattle, WA. https://doi.org/10.1145/3017680.3017755

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. J*ournal of Science Education and Technology*, 25(1), 127–147.

Welcome to Project GUTS | Project GUTS. (n.d.). Retrieved March 15, 2019, from http://www.projectguts.com/

Xue, Y., & Larson, R. C. (2015). STEM crisis or STEM surplus? Yes and yes. *Monthly Labor Review*, 2015. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5800410/

PHOTO RIGHTS AND LICENSING

Appendix A: Research Method

The authors engaged in two approaches to develop this paper - collaborative knowledge building was utilized to develop the CSed Visions framework, and purposive identification of educational programs or curricula that indexed rationales found in that framework was used to develop the “CSed Visions in Action” examples.

CS Visions framework development. In order to surface underlying arguments for universal computer science education, the authors purposefully sampled 24 New York City-based stakeholders in the CS education community–14 females and 10 males from diverse racial and ethnic backgrounds, who held positions in informal digital media education, research, program design, technology development, and education policy and administration. During a 3 hour workshop, participants engaged in a participatory knowledge building (PKB) process (Santo, Ching, Peppler & Hoadley, 2017) facilitated by the authors. This workshop aimed to solicit diverse rationales for CS education. Participants engaged in a range of activities that involved articulating existing arguments and rationales for CS education, working in groups to refine distinct themes that were emerging, noting what kinds of outcomes might relate to different rationales and providing examples of existing learning experiences (e.g., afterschool programs, classroom units) that they felt reflected elements of the arguments.

The CS Visions framework was then derived from an iterative analysis of the arguments participants articulated during the workshop. 161 arguments around the purposes of CS education were identified in total, and were augmented with the content surfaced on participants’ collaborative e-notepads, and the authors’ meeting fieldnotes, sense-making discussion, and analytic memos. The draft framework was presented back during a member checking session where another 16 formal educators repeated the activity, and a session attended by 26 stakeholders (including 8 who had attended the initial workshop). Following this the authors revisited and revised the framework. The framework development methodology is detailed more extensively in the research paper where it was first published (Vogel, Santo & Ching, 2017).

Purposive identification of CS education programs indexing different visions. With an eye towards considering relationships between visions and pedagogies, the authors reviewed data from the workshop described above to find examples offered by participants of various educational programs, tools or curricula that related to various rationales linked to the framework. These examples were used as a starting point to develop the “visions in action” examples highlighted in this paper, and were augmented with additional curricula and programs that the authors were familiar with from past partnerships and participation in local and national CS ed networks. Programs were identified and included based on their capacity to highlight how various rationales for computer science education might result in distinctive pedagogical approaches.

Appendix B: Contributors

The CS Visions Framework was developed through a participatory knowledge building process with New York City-based educators in 2016. The following individuals participated in workshops and roundtables, and contributed ideas that were instrumental to the development of this paper. Affiliations noted were those held at the time of the workshops.

Kaho Abe, Eyebeam
Eric Allatta, New York City Department of Education
Brad Ashley, New York City Department of Education
Maor Baror, CSNYC
Greg Benedis-Grab, The Packer Collegiate Institute
Matthew Berland, University of Wisconsin - Madison
Karen Blumberg, The Brearley School
Josué Cardona, Global Kids, Inc
Joliz Cedeño, Code Nation
Francisco Cervantes, ExpandEd Schools and the Department of Education
Kara Chesal, New York City Department of Education
Brian Cohen, Beam Center
Maurya Couvares, Code Nation
Kelly Deckelman-Brandon, Avenues: The World School
Leigh Ann DeLyser, CSNYC
Michael Denton, Code/Interactive
Jim Diamond, Educational Development Center
Kristina Drury, TYTHE Design
Katrina Falkner, University of Adelaide
Rebecca Garcia, Coder Dojo
Leah Gilliam, Girls Who Code
Jeffrey Goett, Uncommon High School
Tashay Green, Code Nation
Gaelen Hadlett, Sunset Sparks
Rob Hemstreet, New York City Department of Education
Chris Hoadley, New York University
Jacques Hoffman, New York City Department of Education
Nathan Holbert, Teachers College, Columbia University
Tali Horowitz, Common Sense Media
Peter Jabloski, Code/Interactive
Calena Jamieson, Black Girls Code
Sarah Judd, Girls Who Code
Hillary Kolos, DreamYard
Nathan Larsen, New York City Department of Education
Marc Lesser, Mouse, Inc.
Diane Levitt, Cornell Tech
Allisyn Levy, BrainPop
Deidre Levy, New York City Department of Education
Bruce Lincoln, Silicon Harlem
Jasmine Ma, New York University
Meghan McDermott, Mozilla Foundation
Joe Melendez, New York City Department of Education
Ariam Mogos, Global Kids, Inc
Doug Moore, New York Hall of Science
Shane Morris, Code Nation
Saranii Muller, New York City Department of Education
Ramsey Nasser, Eyebeam
Anthony Negron, New York Hall of Science
Tom O'Connell, Code/Interactive
Aankit Patel, New York City Department of Education
Kelly Powers, New York City Department of Education
Michael Preston, CSNYC
Meg Ray, Codesters and Cornell Tech
Robert Robinson, The Graduate Center, City University of New York
Liza Stark, LittleBits
Armando Somoza, Urban Arts Partnership
Mary Ellen Sullivan, CSNYC
Hana Sun, Mozilla Foundation
Emily Tompsett, Montclair Kimberly Academy
Rebecca Vivian, University of Adelaide
Stephanie Yang, Teachers College, Columbia University

For more information about this work, visit visions.csforall.org. To become a member of CSforALL, visit csforall.org/become_a_member.

Back cover, clockwise from top left: courtesy of Global Kids, photo by Melissa Salama; courtesy of New York City Department of Education Computer Science for All; courtesy of Global Kids, photo by Jackson Gomes; courtesy of New York City Department of Education Computer Science for All; courtesy of Global Kids.

Front cover, clockwise from top left: courtesy of New York City Department of Education Computer Science for All; courtesy of New York City Department of Education Computer Science for All; courtesy of Global Kids, photo by Man Cheung; courtesy of Global Kids, photo by Jackson Gomes.