



A Revaluation of Computational Thinking in K–12 Education: Moving Toward Computational Literacies

Yasmin B. Kafai¹  and Chris Proctor² 

Over the past decade, initiatives around the world have introduced computing into K–12 education under the umbrella of computational thinking. While initial implementations focused on skills and knowledge for college and career readiness, more recent framings include situated computational thinking (identity, participation, creative expression) and critical computational thinking (political and ethical impacts of computing, justice). This expansion reflects a revaluation of what it means for learners to be computationally-literate in the 21st century. We review the current landscape of K–12 computing education, discuss interactions between different framings of computational thinking, and consider how an encompassing framework of computational literacies clarifies the importance of computing for broader K–12 educational priorities as well as key unresolved issues.

Keywords: computers and learning; computer science education; critical theory; identity; literacy

In the January 2013 issue of *Educational Researcher*, Grover and Pea reviewed the state of computational thinking within K–12 education, noting that it was “an idea whose time had come” (p. 38). Computational thinking—defined by Jeanette Wing (2006) as “involving solving problems, designing systems and understanding human behavior that draws on concepts fundamental to computing” (p. 33)—was described as a key motivator for bringing computer science (CS) back into schools. Computational thinking’s moment most certainly came just a month later when the video “What Most Schools Don’t Teach” (2013) was released on YouTube, informing millions of viewers that today’s children need to learn about CS. A large cast of celebrities, among them Microsoft founder Bill Gates, basketball star Chris Bosch, and rock musician will.i.am promoted the cause. In the video, a quote by the late Steve Jobs, cofounder of Apple, made the connection to computational thinking by demanding that “everybody in this country should learn how to program a computer . . . because it teaches you how to think.” Later that same year the Hour of Code was launched during Computer Science Education Week, giving millions of K–12 students their first taste of programming. Eight years later, over one billion students around the world have participated in this annual event.

These are surprising developments given earlier reports about the dismal state of K–12 CS education, which documented

an absence of courses and teachers in schools, and an alarming lack of diversity in the field (Wilson et al., 2010). With CS as an elective not counting toward high school graduation, few students were taking AP courses when they were offered at all (Margolis et al., 2012). Neither teacher preparation programs nor certifications were available, and school districts had no frameworks or standards in place. Fast forward to 2021, the number of students taking AP courses and participating in the exam has significantly increased (though still not reaching participation comparable to other popular AP topics such as statistics), teacher certification standards and programs are being established, and many school districts now count CS courses as graduation requirements. By all measures, this is significant progress even though much work still remains to be done in making CS an integral part of the K–12 education system (e.g., DeLyser et al., 2018).

Indeed, computational thinking has become part of a global effort under which the introduction of CS education in K–12 is promoted under different directives, most prominently to maintain each country’s economic competitiveness (Hsu et al., 2019) but also to develop a new cross-disciplinary literacy: “. . . to reading, writing and arithmetic, we should add computational

¹University of Pennsylvania, Philadelphia, PA

²University at Buffalo, SUNY, Buffalo, NY

thinking to every child's analytical ability" (Wing, 2006, p. 33). The latter directive, becoming computationally literate, is not just about learning technical skills such as the pragmatics of reading and writing text (or code) but also about how these skills are recognized and valued within particular personal and political dimensions (Scribner, 1984). Thus adding computational thinking to the canon of language, mathematics, and science literacies (Guzdial, 2019) means expanding beyond narrow framings which emphasize solving problems with computers toward an understanding of the values, biases, and histories embedded in computational technologies and cultures which run on computers. Such an expanded framing of computational thinking will include pragmatic, sociocultural, and political dimensions in order to address critical aspects of inequities caused or exacerbated by the societal impact of computing (e.g., Margolis et al., 2008/2017) and the growing prominence of CS as an academic field.

Incorporating broader definitions of computational thinking into the goals of primary and secondary education requires a clear articulation of what this would look like for the individual and for society. CS education is still an emerging field at the K–12 level, and the diversity of K–12 education, both in terms of who participates and in terms of the priorities and positionality of stakeholders, demands careful attention to theoretical framings. We situate current efforts within theoretical educational frameworks that consider learning and teaching. While Grover and Pea (2013) framed computational thinking in mostly cognitive terms, they already anticipated the need for expanded perspectives. We illustrate how these expanded framings can help us understand in which ways computational thinking can become a centerpiece in promoting computational literacies relevant to K–12 education. Like its STEM counterparts mathematics and science, CS education is embarking on an ambitious agenda and needs to consider not just one but multiple visions for K–12 learners (Vogel et al., 2017). In the conclusion, we address critical issues of who and what we have in mind when computational thinking is promoted, and how we can ensure that the purposes of computational literacies include not only an understanding of key ideas and practices but also its socially responsible and critical uses.

Three Framings of Computational Thinking

In many of today's national initiatives, standards, curricula, and courses, computational thinking has been adopted as a general-purpose skill that all students need to learn. Much discussion in CS (e.g., Tedre & Denning, 2016) and K–12 education (e.g., National Research Council, 2010) has focused on whether computational thinking should be defined as a skill for general problem solving or whether it is mostly applicable to problems within CS (Curzon et al., 2019). Others collapse the distinction by proposing to make computing more generally accessible for solving everyday problems. Regardless of how broadly applicable particular skills may be, the goal of competency in solving problems with computers is presented as an uncontroversial good. We argue that not enough attention has been paid to how students' learning of and with computation is framed through lenses of different learning theories. Any kind of learning is framed

through metaphors (Sfard, 1998) each of which carry tacit assumptions and beliefs not only regarding how people learn best but also about the nature of knowledge and its uses. These metaphors guide how we set educational priorities, how we teach, and how we assess learning.

A recent paper presented at the International Computing Research Education Conference (Kafai, Proctor, & Lui, 2019) identified three framings of computational thinking to help organize the theory space of desired outcomes in computational thinking, both within K–12 CS and in K–12 education more broadly. Building on Sfard (1998), the paper distinguished between (1) cognitive framings viewing learning as the acquisition of knowledge and skills and emphasizing preparation for future careers, (2) situated framings viewing learning as identity formation through participation in disciplinary practices and emphasizing creative expression and social engagement in digital media, and (3) critical framings viewing learning as developing an understanding of how realities are shaped and emphasizing strategies for resisting marginalization and oppression.

Articulating these theoretical framings reveals that thinking computationally has a much broader scope than what Grover and Pea (2013) found in their initial review of the state of the field. Studies conducted under a cognitive framing focus on students' misconceptions about programming concepts, students' various challenges with debugging computer programs, or students' attitudes toward programming (Buitrago Flórez et al., 2017). The majority of research investigating students' computational thinking takes a cognitive stance employing theoretical framings prevalent in the 1980s (Tenenberg & Knobelsdorf, 2014). A large number of related efforts also promote the integration of computational thinking into STEM disciplines (e.g., Weintrop et al., 2016). Many national standards (e.g., K–12 CS Framework, 2016) and curricula such as Code.org's (2020) "Computer Science Discoveries" have adopted this cognitive framing and mapped out learning progressions and pathways for how students should develop computational thinking, starting as early as kindergarten.

One key limitation of a cognitive framing is that student learning outcomes are seen mostly in individualistic terms, paying little attention to how learning is embedded in social and cultural contexts (Lave & Wenger, 1991). Even when studies include collaborative arrangements such as the popular pair programming, the focus is on coordinating interactions between individual learners (Campe et al., 2020). But in order to address social and cultural contexts in learning and teaching CS, a discipline that historically has marginalized women and students of color (Margolis et al., 2008/2017), there is also need

to develop theoretical knowledge about how we can design learning environments to support youth from nondominant communities—who learn and develop along racialized, gendered, and class-influenced learning pathways—in their disciplinary identity development and their navigation of the social worlds in which they currently participate and those they desire to join. (Bell et al., 2017, p. 369)

Situated framings of computational thinking understand learning in terms of identity, practice, and participation (Sfard,

1998) and see computing as a vehicle for personal expression and connecting with others alongside and intersecting a plurality of other literacy practices. Here applications have focused on connecting the learning of programming with students' prior interests, including music, storytelling, or, most often, video games. Studies have also examined participation in programming communities resulting in tens of thousands of digital artifacts created by children in the Scratch online community, referencing popular commercial game franchises and narratives (Kafai & Burke, 2014). Example curricula include the Creative Computing Curriculum (Brennan et al., 2011) which situates students' introduction to computational thinking through a variety of game design and storytelling activities or Stitching the Loop (Kafai, Fields, et al., 2019) activities that engage high school students in crafting and coding personalized electronic textiles. Efforts in this direction first emerged in community technology centers and online communities but have now also been integrated in formal schooling contexts (see Kafai, Fields, et al., 2019).

A key feature of the situated framing is the recognition of authentic learning practices and the realization that learning means becoming a member of a community of practice with shared goals and values (Sfard, 1998). Learning computational concepts and practices are often contextualized within the design of digital applications shared with authentic audiences in person or over social networks. The situated framing has been adopted and led by sociocultural researchers from the learning sciences with a recognition that inequitable access to opportunities to participate, develop interest, and have one's identities supported are a root cause of the lack of diversity in CS. But truly supporting marginalized students requires helping them understand and contest the forces which exclude them. Some researchers (e.g., Vakil, 2018) argue that situated computational thinking does not go far enough in confronting forces such as racism and sexism.

A critical framing of computational thinking contextualizes cognitive and situated practices within broader cultural formations such as race, gender, class, and language. One mode of criticality, "computing in the world," focuses on understanding the role of computational infrastructure in society, and particularly its role in reproducing oppression. This includes, for instance, how implicit racial bias is embedded in crime-prediction software used by police, or how mass surveillance and social credit ranking algorithms consolidate state power. The goals are for students to use computing for social good, to analyze computational aspects of social issues, and to commit to future ethical computational practice (e.g., Mozilla's [2021] *Teaching Responsible Computing Playbook*). A second mode of criticality focuses on the inverse: "the world in computing." In this approach, computing is understood as a site of social activity (rather than as a tool for influencing society) where social hierarchies are reproduced in new forms. This approach might focus on sexism in gaming communities and dating apps, or on how computing education and computing careers are themselves sites of racial violence (Rankin et al., 2021). Recent work puts these two approaches in dialogue with one another, showing how critical transformation of computing classrooms allows them to catalyze student activism more broadly (Ryoo et al., 2020).

While some critical computational thinking work smoothly expands the inclusivity of cognitive and situated approaches, critical framings also tend to challenge the assumptions of mainstream approaches. For instance, Williamson (2016) argues that computational thinking is part of an ongoing corporate incursion into public education with the political effect of producing computationally governable citizens willing to accept computational framings of social problems. Through this lens, marginalized populations' purported lack of interest in computing may also represent a strategy of refusal. Critical approaches to computational thinking have also drawn on critical race theory to sharpen critiques of diversity and inclusion rhetoric which ignores underlying injustices (Scott & White, 2013).

In reevaluating the state of computational thinking, we observe substantial growth in situated framings of computational thinking anticipated by Grover and Pea (2013), as well as growth in critical framings not foreseen by the previous review. Like others (Tenenberg & Knobelsdorf, 2014), we find that the cognitive framing of computational thinking has been the dominant and largely unquestioned paradigm in the most recent wave of K-12 CS education initiatives. Rather than considering the different framings in isolation—as we did for analytical purposes in our review—we should begin integrating them to develop a fuller and richer version of computational thinking (Kafai, Proctor, et al., 2019). We are not alone in the quest for an expanded, integrated vision for CS education. Recent calls have emerged to change and broaden its aims "from the computer to programming, algorithms and information, as well as to the organizational, social and cultural environment of computer systems" (Tedre et al., 2018, p. 158). This new and expanded perspective on computational thinking directs our attention to its larger role and purpose within K-12 education, and the question of what we mean by computational literacies.

Moving Toward Computational Literacies

In the previous section, we showed that multiple framings of computational thinking are needed to describe the variety of learning goals included under computational thinking. Rather than overloading the concept of computational thinking with multiple meanings, we join an emerging movement of scholars adopting the framework of computational literacies. Our use of the plural *literacies* follows the New London Group's (1996) recognition of multiple literacies, distinguished by different media technologies and different cultural practices. We also draw on *literacies* as discussed in the learning sciences, with particular attention to how technology shapes cognition and communication. For example, diSessa's (2001) analysis of computational literacies identified cognitive and social aspects of *material intelligence*, or thinking "achieved cooperatively with external materials" (p. 5). We would bring together and extend these definitions to include critical perspectives which emphasize the ways reading and writing practices are located within broader power relations and how literacy functions as an instrument of power. Building on work by Barton and Hamilton (1998), Gee (2004), Lee and Garcia (2014), and Jacob and Warschauer (2018), we define computational literacies as a set of practices situated in a sociocultural context which utilize external computational

media to support cognition and communication. Computational literacies encompass phenomena at scales from the individual to the societal, as well as connections between these phenomena and the media which supports and shapes them.

Using the lens of computational literacies clarifies three core questions about the relationship between CS and K–12 education. First is the question of *whom*: Who should learn CS? The answer appears straightforward: CS is for everyone. Wing’s (2006) assertion that “everyone, not just computer scientists, would be eager to learn and use” computational thinking is echoed in the K–12 CS Framework’s argument that CS is “invaluable to all parts of life and important beyond ensuring that we have enough skilled technology workers” (p. 33). While technology careers are one possible endpoint for K–12 CS education, they should not be the only option. Instead we also need to include vocational training, civic engagement, and creative expression as possible options (Tissenbaum et al., 2021). For that reason the far-from-universal rates of CS participation are problematic and inequitable. From the perspective of computational literacies, the cognitive and situated practices which constitute CS are endemic to a particular, highly privileged community. These practices have been phenomenally successful, but it does not necessarily follow that the same practices will serve other communities, with their own goals, in the same way. Does “CS for all” imply that everybody ought to change to accommodate CS, or that CS ought to change to accommodate everybody? In our view, there are parallels to the claim that “Standard English is for everyone,” both in terms of why the claim is problematic and how we might productively advance a discussion of educational priorities.

Paradigms such as the “digital divide” position nondominant youth in particular as either being literate or illiterate in the use of digital tools. The technology industry has perpetuated views organized around the “haves and have nots”—with “access” as a proxy for a “culture of poverty.” This has spawned new media and digital literacy initiatives (e.g., coding academies) that have shaped the role of emergent digital technologies in youths’ learning both in and out of school (Nasir & Vakil, 2017). Such initiatives are organized around a form of technological determinism that delimits the possibilities of tools and relies on practices that often conflict with youths’ personal, sociocultural learning interests, as well as with their innovative, ingenious (McDermott & Raley, 2011), and sometimes subversive uses of technologies. Thus, we are interested in how youth from nondominant communities employ digital tools to navigate these tensions and to develop agentic practices that in turn signify historical action.

Second is the question of *what*. Grover and Pea (2013) noted the unresolved definitional quagmires that have plagued computational thinking since its inception. While debate over the definition of computational thinking remains alive and well, major stakeholders have made a strategic priority of settling on a single definition which can be the basis of standards for curriculum and teacher certification (Barr & Stephenson, 2011). This work culminated in the K–12 CS Framework (2016), a consensus document which aims to align standards and curricula while sidestepping some of the core definitional questions about computational thinking.

At first sight, a universal, unambiguous definition of an academic discipline seems a laudable objective, mirroring what other STEM disciplines promote in their frameworks and

standards for K–12 education. However, defining literacies means defining who is literate and which practices (always bound up in cultures, places, and identities) are legitimate. Such definitions could be helpful in constructing pathways to educational opportunity, but they could also contribute to blaming victims for their own marginalization. In our view, the question of what to teach in K–12 CS need not have a single answer, but could instead have many answers grounded in the computational literacy practices of diverse communities and cultures.

Finally, computational literacies can help clarify a third question of *how*. How should we teach computational thinking? Should it be taught as a stand-alone topic or integrated within other K–12 classes? Despite considerable evidence that learning computational thinking does little for students’ general problem-solving abilities, we noted above that decontextualized curricula and assessments dominate current K–12 CS education initiatives. These approaches suggest the priority is scalability and evaluation, with the assumption that once students have been taught the foundational skills and knowledge, they will naturally put them to use in critical and responsible ways. However, this approach risks a one-directional integration of computational thinking into existing literacies: tailoring instruction to make it relevant to diverse cultures and identities (as suggested by the metaphor of a “leaky pipeline” of STEM education) without also locating educational goals in those cultures and identities, and then asking how (or if) computation might contribute. The latter approach implies a collaborative process of deciding what matters and developing assessments aligned with these values and understandings. The “Who,” “What,” and “How” of CS education are challenging questions to address. The introduction of computational literacies allows us to draw on the rich traditions of new literacies and critical literacies scholarship while also posing significant challenges in development of assessments. This essential work is required for any expansion into K–12 education.

Priorities for K–12 Inquiries

Grover and Pea (2013) concluded their review of computational thinking by calling for consideration of broader theoretical perspectives and for research on K–12 CS implementations. The intervening years have resulted in substantial research along these lines, much of which contributed to the fragmentation of computational thinking into cognitive, situated, and critical framings (see Kafai, Proctor, & Lui, 2019). In its initial conception, learning computational thinking was aimed at facilitating interactions with emerging digital technologies, mostly present at the workplace. Today, these digital technologies are embedded in every aspect of our professional, public, and private lives; their importance demands a broader conceptualization of computational thinking into literacies. For this reason, we highlighted several critical issues facing K–12 CS resulting in a move toward literacies spanning multiple scales. We close by suggesting further areas of needed research and development.

We call for more research in understanding of how K–12 students engage with various computational concepts and practices and possible trajectories of how they deepen their understanding, keeping in mind alternate endpoints of what it means to be

computationally literate. Such investigations should focus not just on the pathways and challenges in developing students' computational skills and concepts (as the current K–12 CS Framework does) but also be inclusive of students' identities and their respective communities. The identities and communities to which such learning might be relevant and the ends to which learning might be put must be constantly in view. A computational literacies perspective would cast a much wider net in how we come to understand K–12 students' learning and participation.

We also call for the development of transformative pedagogies to address political and ethical implications in teaching CS, as well as the inadequacy of simply adding an ethics module to the curriculum. Instead, we need to investigate how pedagogy can be suffused with a critical ethos even when teaching specific technical content. Moreover, we need to better understand the obstacles known to interfere with introducing socially responsible applications or widespread interest and access. Here a focus on computational literacies is useful for framing the multiple, simultaneous aims of pedagogy. Computational literacies recognize the positionality of computational practices, offering tools for critical action and recognizing the role of myriad identities and cultures in developing a rich diversity of computational practice.

Accompanying these calls is a necessity to make headway in assessing computational literacies at different scales. While schools are accountable for individual student outcomes, they also need to ensure that the kinds of outcomes they measure and how they measure them are aligned with the cultures and communities they serve. Rather than positioning the cultures and political priorities of a school's community as external to learning or as resources to be appropriated to catalyze learning, a computational literacies approach recognizes a school community as stakeholders in shaping what it means to practice CS at that school. In practice, equitably including stakeholders in defining and designing CS requires sustained outreach (Proctor et al., 2019), but we imagine many articulations of CS flourishing at different schools, each reflecting the voices of its community.

Finally, we need to tackle the key challenge of including computing education in teacher education programs (DeLyser et al., 2018) as part of the larger education system. Current efforts have focused on getting new teachers into CS or integrating computing within STEM topics. While each of these efforts is important, we need to develop a better understanding of what content knowledge and skills K–12 teachers actually need to have, what equitable teaching practices look, and how teachers can address critical dimensions of CS. We know that in addition to deepening teachers' pedagogical practices and content knowledge, continued professional development will be instrumental for retaining them in the teaching profession (Ingersoll & Smith, 2003).

In directing our attention to the multiple scales at which computing shapes our world, computational literacies set the bar high for the responsibilities of education researchers, practitioners, and designers. If K–12 CS education is to reach its potential for individual opportunity and social transformation, the broad goals articulated by computational literacies must be followed by implementations and assessments which recognize its complexities. It is our hope that reorienting computational thinking around computational literacies is a step toward clarifying what it would mean to realize the potential and necessity of K–12 CS education in full.

ORCID IDs

Yasmin B. Kafai  <https://orcid.org/0000-0003-4018-0491>

Chris Proctor  <https://orcid.org/0000-0003-3492-9590>

NOTE

The writing of this article was supported by a grant from the National Science Foundation to Yasmin Kafai (#1742140). Any opinions, findings, and conclusions or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of NSF, the University of Pennsylvania, or University at Buffalo, SUNY. The authors have no known conflict of interest to disclose.

REFERENCES

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K–12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48–54. <https://doi.org/10.1145/1929887.1929905>

Barton, D., & Hamilton, M. (2012). *Local literacies: Reading and writing in one community*. Routledge.

Bell, P., Van Horne, K., & Cheng, B. H. (2017). Designing learning environments for equitable disciplinary identification. *Journal of the Learning Sciences*, 26(3), 367–375. <https://doi.org/10.1080/10508406.2017.1336021>

Brennan, K., Balch, C., & Chung, M. (2011). *Creative computing curriculum*. <https://scratch.mit.edu/guide>

Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., & Danies, G. (2017). Changing a generation's way of thinking: teaching computational thinking through programming. *Review of Educational Research*, 87(4), 834–860. <https://doi.org/10.3102/0034654317710096>

Campe, S., Denner, J., Green, E., & Torres, D. (2020). Pair programming in middle school: Variations in interactions and behaviors. *Computer Science Education*, 30(1), 22–46. <https://doi.org/10.1080/08993408.2019.1648119>

Code.org. (2020). *Computer science discoveries*. <https://code.org/educate/csd>

Curzon, P., Bell, T., Waite, J., & Dorling, M. (2019). Computational thinking. In S. A. Fincher & A. Robbins (Eds.), *Cambridge handbook of computer science education research* (pp. 513–546). Cambridge University Press.

DeLyser, L. A., Goode, J., Guzdial, M., Kafai, Y., & Yadav, A. (2018). *Priming the computer science teacher education pump: Finding a home for computer science education in school of education*. CS4All. <https://www.computingteacher.org>

diSessa, A. A. (2001). *Changing minds: Computers, learning, and literacy*. MIT Press.

Gee, J. P. (2004). Learning language as a matter of learning social languages within discourses. In M. R. Hawkins (Ed.), *Language learning and teacher education: A sociocultural approach* (pp. 13–32). Multilingual Matters.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43. <https://doi.org/10.3102/0013189X12463051>

Guzdial, M. (2019). Computing education as a foundation for 21st century literacy. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 502–503). ACM. <https://doi.org/10.1145/3287324.3290953>

Hsu, Y.-C., Irie, N. R., & Ching, Y.-H. (2019). Computational thinking educational policy initiatives (CTEPI) across the globe. *TechTrends*, 63(3), 260–270. <https://doi.org/10.1007/s11528-019-00384-4>

Ingersoll, R., & Smith, T. (2003). The wrong solution to the teacher shortage. *Educational Leadership*, 60(8), 30–33.

Jacob, S. R., & Warschauer, M. (2018). Computational thinking and literacy. *Journal of Computer Science Integration*, 1(1), 1. <https://doi.org/10.26716/jcsi.2018.01.1.1>

K-12 Computer Science Framework. (2016).

Kafai, Y. B., & Burke, Q. (2014). *Connected code: Why children need to learn programming*. MIT Press.

Kafai, Y. B., Fields, D. A., Lui, D. A., Walker, J. T., Shaw, M. S., Jayathirtha, G., Nakajima, T. M., Goode, J., & Giang, M. T. (2019). Stitching the loop with electronic textiles: Promoting equity in high school students' competencies and perceptions of computer science. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 1176–1182). <https://doi.org/10.1145/3287324.3287426>

Kafai, Y., Proctor, C., & Lui, D. (2019). From theory bias to theory dialogue: Embracing cognitive, situated and critical framings of computational thinking for K-12 CS education. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (pp. 101–109). ACM. <https://doi.org/10.1145/3291279.3339400>

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge University Press.

Lee, C. H., & Garcia, A. D. (2014). "I want them to feel the fear . . .": Critical computational literacy as the new multimodal composition. In R. E. Ferdig & K. E. Pytash (Eds.), *Exploring multimodal composition and digital writing* (pp. 364–378). Information Science Reference.

Margolis, J., Estrella, R., Goode, J., Holme, J. J., & Nao, K. (2017). *Stuck in the shallow end: Education, race, and computing*. MIT Press. (Original work published 2008)

Margolis, J., Ryoo, J. J., Sandoval, C. D. M., Lee, C., Goode, J., & Chapman, G. (2012). Beyond Access: Broadening Participation in High School Computer Science. *ACM Inroads*, 3(4), 72–78. <https://doi.org/10.1145/2381083.2381102>

McDermott, R., & Raley, J. (2011). Looking closely: Toward a natural history of human ingenuity. In E. Margolis & L. Pauwels, *The SAGE handbook of visual research methods* (pp. 372–391). Sage. <https://doi.org/10.4135/9781446268278.n20>

Mozilla. (2021). *Teaching responsible computing playbook*. K. Pham & A. Ruda (Eds.) <https://foundation.mozilla.org/en/what-we-fund/awards/teaching-responsible-computing-playbook/>

Nasir, N. S., & Vakil, S. (2017). STEM-focused academies in urban schools: Tensions and possibilities. *Journal of the Learning Sciences*, 26(3), 376–406. <https://doi.org/10.1080/10508406.2017.1314215>

National Research Council. (2010). *Report of a workshop on the scope and nature of computational thinking*. National Academies Press.

The New London Group. (1996). A pedagogy of multiliteracies: Designing social futures. *Harvard Educational Review*, 66(1), 60–93.

Proctor, C., Bigman, M., & Blikstein, P. (2019). Defining and designing computer science education in a K12 public school district. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 314–320). ACM. <https://doi.org/10.1145/3287324.3287440>

Rankin, Y. A., Thomas, J. O., & Erete, S. (2021). Real talk: Saturated sites of violence in CS education. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (pp. 802–808). ACM. <https://doi.org/10.1145/3408877.3432432>

Ryoo, J. J., Tanksley, T., Estrada, C., & Margolis, J. (2020). Take space, make space: How students use computer science to disrupt and resist marginalization in schools. *Computer Science Education*, 30(3), 337–361. <https://doi.org/10.1080/08993408.2020.1805284>

Scott, K.A., & White, M.A. (2013). COMPUGIRLS' standpoint: Culturally responsive computing and its effect on girls of color. *Urban Education*, 48(5), 657–681. <https://doi.org/10.1177/0042085913491219>

Scribner, S. (1984). Literacy in three metaphors. *American Journal of Education*, 93(1), 6–21. <https://doi.org/10.1086/443783>

Sfard, A. (1998). On two metaphors for learning and the dangers of choosing just one. *Educational Researcher*, 27(2), 4–13. <https://doi.org/10.3102/0013189X027002004>

Tedre, M., & Denning, P. J. (2016). The long quest for computational thinking. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research—Koli Calling '16* (pp. 120–129). ACM. <https://doi.org/10.1145/2999541.2999542>

Tedre, M., Simon, B., & Malmi, L. (2018). Changing aims of computing education: A historical survey. *Computer Science Education*, 28(2), 158–186. <https://doi.org/10.1080/08993408.2018.1486624>

Tenenberg, J., & Knobelsdorf, M. (2014). Out of our minds: A review of sociocultural cognition theory. *Computer Science Education*, 24(1), 1–24. <https://doi.org/10.1080/08993408.2013.869396>

Tissenbaum, M., Weintrop, D., Holbert, N., & Clegg, T. (2021). The case for alternative endpoints in computing education. *British Journal of Educational Technology*, 52(3), 1164–1177. <https://doi.org/10.1111/bjet.13072>

Vakil, S. (2018). Ethics, identity, and political vision: Toward a justice-centered approach to equity in computer science education. *Harvard Educational Review*, 88(1), 26–52. <https://doi.org/10.17763/1943-5045-88.1.26>

Vogel, S., Santo, R., & Ching, D. (2017). Visions of computer science education: Unpacking arguments for and projected impacts of CS4All initiatives. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education—SIGCSE '17* (pp. 609–614). ACM. <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. *Journal of Science Education and Technology*, 25(1), 127–147. <https://doi.org/10.1007/s10956-015-9581-5>

What most schools don't teach. (2013). *YouTube*. <https://www.youtube.com/watch?v=nKlu9yen5nc>

Williamson, B. (2016). Political computational thinking: Policy networks, digital governance and "learning to code." *Critical Policy Studies*, 10(1), 39–58. <https://doi.org/10.1080/19460171.2015.1052003>

Wilson, C., Sudol, L. A., Stephenson, C., & Stehlík, M. (2010). *Running on empty: The failure to teach K-12 computer science in the digital age*. ACM. <https://doi.org/10.1145/3414583>

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>

AUTHORS

YASMIN B. KAFAI, EdD, is Lori and Michael Milken President's Distinguished Professor at the University of Pennsylvania, Graduate School of Education, 3700 Walnut Street, Philadelphia, PA 19104; kafai@upenn.edu. Her research focuses on the development and study of tools, communities, and materials promoting computational participation, crafting, and creativity across K–16.

CHRIS PROCTOR, PhD, is an assistant professor of learning sciences at the University at Buffalo, SUNY, 510 Baldy Hall, Buffalo, NY, 14228; chrsp@buffalo.edu. He studies K–12 computer science education through the disciplinary lenses of learning sciences and new literacies.

Manuscript received October 27, 2020
 Revised May 25, 2021, and August 19, 2021
 Accepted October 1, 2021