How Economically Marginalized Adolescents of Color Negotiate Critical Pedagogy in a Computing Classroom

JEAN SALAC, University of Washington, Seattle, WA, USA
LENA ARMSTRONG, University of Pennsylvania, Philadelphia, PA, USA
F. MEGUMI KIVUVA, JAYNE EVERSON, and AMY J. KO, University of Washington, Seattle, WA, USA

Background and Context: With the growing movement to adopt critical framings of computing, scholars have worked to reframe computing education from the narrow development of programming skills to skills in identifying and resisting oppressive structures in computing. However, we have little guidance on how these framings may manifest in classroom practice.

Objectives: To better understand the processes and practice of critical pedagogy in a computing classrooms, we taught a critically conscious computing elective within a summer academic program at a northwest United States university targeted at secondary students (ages 14–18) from low-income backgrounds and would be the first in their families to pursue a post-secondary education (i.e., first-generation). We investigated: (1) our participants' initial perceptions of and attitudes toward the benefits and perils of computing, and (2) potential tensions that might emerge when secondary students negotiate the integration of critical pedagogy in a computing classroom.

Methods: We qualitatively coded participant work from a critically conscious computing course within a summer academic program in the United States focused on students from low-income backgrounds or would be the first in their family to pursue a post-secondary education.

Findings: Our participants' initial attitudes toward technology were mostly positive, but exhibited an awareness of its negative impacts on their lives and society. Throughout the course, while participants demonstrated a rich social consciousness around technology, they faced challenges in addressing hegemonic values embedded in their programs, designs, and other classwork.

Implications: Our findings revealed tensions between our participants' computing attitudes, knowledge, self-efficacy, and social consciousness, suggesting pathways for scaffolding the critical examination of technology in secondary education. This study provides insights into the pedagogical content knowledge necessary for critical computing education.

CCS Concepts: • **Social and professional topics** → **Computing education**; **Adolescents**; *Computing literacy*;

Additional Key Words and Phrases: critical pedagogy, adolescents, secondary education

This material is based upon work supported by the National Science Foundation under grants 2127309, 1539179, 1703304, 1836813, 2031265, 2100296, 2122950, 2137834, 2137312, and unrestricted gifts from Microsoft, Adobe, and Google.

Authors' Contact Information: Jean Salac (corresponding author), University of Washington, Seattle, WA, USA; e-mail: salac@uw.edu; Lena Armstrong, University of Pennsylvania, Philadelphia, PA, USA; e-mail: lena318@sas.upenn.edu; F. Megumi Kivuva, University of Washington, Seattle, WA, USA; e-mail: megumik@uw.edu; Jayne Everson, University of Washington, Seattle, WA, USA; e-mail: everjay@cs.washington.edu; Amy J. Ko, University of Washington, Seattle, WA, USA; e-mail: ajko@uw.edu.

ACM 1946-6226/2024/12-ART2
https://doi.org/10.1145/3702330

ACM Reference format:
Jean Salac, Lena Armstrong, F. Megumi Kivuva, Jayne Everson, and Amy J. Ko. 2024. How Economically Marginalized Adolescents of Color Negotiate Critical Pedagogy in a Computing Classroom. *ACM Trans. Comput. Educ.* 25, 1, Article 2 (December 2024), 29 pages.
https://doi.org/10.1145/3702330

1 Introduction

There has been a growing awareness of how technology can reinforce and amplify oppression in society [4, 13, 14] because of implicit harmful and exclusionary values embedded in their design [75]. As a result, there has been an emerging movement in computing education to go beyond teaching students strictly technical content, expanding to also include critical perspectives on computing and society. By teaching critical framings of computing—framings that make explicit the values embedded in the design of computing technologies—students would learn to recognize and resist oppressive power structures in computing, both in scholarly contexts, but also in life, in work, and as citizens [39]. Similar to other fields which have integrated critical pedagogy [63, 70], computing education research offers compelling insights on *what* to teach in critical computing [33, 65, 76], but there is relatively little guidance on *how* these ideas might manifest in classroom practice and what pedagogical content knowledge may be necessary for critical computing instruction.

Central to understanding how such ideas will translate in the classroom is understanding students' existing knowledge, experiences, and relationships around technology, so that we may understand where they are in their sociopolitical development [71–73]. As a goal of a critical framing of computing is for students to develop conceptions of sociopolitical disparities within computing, it is pertinent to understand the ideological orientations toward computing that students enter the classroom with and consequently, how they may shape students' learning of critical computing ideas. Such an understanding would enable teachers, curriculum developers, and other educators to "meet them where they are" [22].

Students' existing knowledge, experiences and relationships around technology will necessarily differ based on their identities and backgrounds. We selected our study population of students who were from low-income backgrounds and would be the first in their families to pursue a post-secondary education (i.e., first-generation) for several reasons. The harmful and exclusionary values implicit in and amplified by the design of technology have disproportionate effects on the most marginalized by society [4, 13, 14]. Yet, in the US, they are among those with the least representation and power in the design and development of said technologies [21]. Upholding the position that technology is never neutral and is always value-laden [75], our rationale in engaging this study population is to provide them avenues to embed and express their values with technology, making space for examination around not only the hegemonic values to avoid but also which values to actively promote in computing [10].

The movement toward more critical computing education builds upon long-standing efforts to educate students on the perils of computing. These have historically come in the form of teaching students on how to recognize and protect themselves from *individual* harm from technology [16, 25, 31, 48], but have recently extended to identifying and acting on more *systemic* harms from technology. Previous scholars have argued for an interrogation of computing education's dominant emphasis on technical skills and competencies [33, 65, 76], with some scholars starting to experiment with such critical framings of computing in instruction [15, 57]. However, like many efforts to integrate critical pedagogy in other subject areas, these recent efforts have primarily focused on exploring content, not necessarily the processes or practices of critical pedagogy in a computing classroom [70]. Investigating the processes of critical pedagogy have yielded insights around the tensions between the practice of critical pedagogy and the real constraints of classrooms [63].

To better understand processes of critical pedagogy in a computing context, we pursued two research questions:

(1) What perceptions of and attitudes toward computing might students from low-income and first-generation backgrounds bring into critical computing classrooms?
(2) What tensions might emerge when integrating critical pedagogy into a secondary computing classroom context?

We implemented our study in a course based on the secondary education methods in *Critically Conscious Computing* [38] within a summer academic program hosted by a university in northwest US. Because our knowledge of how critical computing pedagogy may manifest in practice is nascent, our objective was to characterize the perceptions, attitudes, and learning experiences of our study population, not to make generalizable or causal claims beyond our context. Given this objective, we opted for a combination of inductive and deductive qualitative coding of classwork. Additionally, we posit that our context, a summer academic program for secondary students (ages 15–17) from low-income backgrounds and would be the first-generation university students, was well-aligned for the formative nature of study for several reasons. First, we were able to draw from an ethnically and linguistically diverse pool of students, a population of students less represented in computing education opportunities, let alone computing education research studies [23]. Second, the context of a summer program as opposed to during the academic year allowed us more flexibility in instructional design and content. Third, our positionality as university researchers shielded us from the risks of teaching critical content, which was especially important amidst the hostile sociopolitical climate teachers are facing in US schools [68].

Through this study, we made two contributions. First, we provide key insights into the possible prior knowledge and experiences with the benefits and perils of computing that students may bring to critical computing instruction, suggesting potential entryways that educators can leverage for critical examination of technology in the classroom. Second, we identified tensions between our participants' perceptions of and attitudes toward computing, their computing self-efficacy and knowledge, and social consciousness when learning computing with a critical lens. These tensions suggest the need for scaffolding in emergent critical computing pedagogy, a promising avenue for investigating pedagogical content knowledge for this framing of computing.

2 Theoretical Framework

Discourse on critical consciousness and critical pedagogy grounded our instructional approach in this study. We therefore draw upon the literature around ideological orientations around computing to make explicit our position toward computing and the position imbued in our pedagogy. Lastly, narrative identity [46] guided our methodological choices to leverage storytelling and speculative fiction.

2.1 Critical Consciousness and Sociopolitical Development

Critical consciousness is an education concept that emphasizes (1) the development of a deep understanding of the world and reality, enabling the awareness of sociopolitical discrepancies, and (2) activism against the oppressive structures that are elucidated by that understanding [17].

In the decades since Freire proposed this idea, scholars have theorized and investigated *how* learners develop critical consciousness. Ladson-Billings suggested that culturally relevant teaching can facilitate learners in developing critical consciousness [40, 41]. Watts et al. further theorized on the development of critical consciousness, proposing the theory of sociopolitical development. This theory delineates how marginalized youth progress through five distinct developmental stages toward critical consciousness: acritical, adaptive, pre-critical, critical, and liberation [71–73]. In the *acritical* stage, youth are oblivious to systemic oppression and can dismiss understandings of marginalization on the basis of identity, such as gender or race. In the *adaptive* stage, through involvement in critical reflection and action, youth learn to recognize systems of oppression but perceive them as unchangeable and adapt to them instead. In the *pre-critical* stage, youth begin to question the value of adapting oppressive system. In the *critical* stage, youth see the need for social-change efforts, and lastly, in the *liberation* stage, youth regularly engage in social action and community development to resist oppressive systems. Like many theories, this theory of sociopolitical development can oversimplify and abstract away nuance [62]. In reality, people are not guaranteed to linearly progress through these stages and their sociopolitical development may vary with context. Understanding its limitations, we drew from this theory, as one tool but not the *only* tool, to help us understand our participants' development of critical consciousness around computing specifically, one of the goals of our class.

Critics of the discourse around critical consciousness have challenged the over-emphasis on awareness-raising at the expense of action against oppressive structures. In the terminology of sociopolitical development theory, the discourse has focused too much on the adaptive and critical stages, taking youths' progression to the liberation stage as a given. For instance, William Paris argued that consciousness is not simply awareness, but instead is justification and prediction [51]. People justify and predict the outcomes of their interactions with social environments and decide what is possible or impossible in their environment. Therefore, when we observe people's behaviors in certain environment, we should be careful when attributing specific ideologies or "awareness" as explanatory. Rather, we should consider the incentive structures of people's environment and how they support or prevent the attainment of specific needs. Paris highlighted a substantive distinction between developing the knowledge of sociopolitical disparities ("awareness-raising") and the knowledge of how to address them, and advocates for further research into the latter. Grounded in the discourse and debates around critical consciousness, our study sought to understand how adolescents may not only develop a critical consciousness around and within a computing educational context, but also begin to act on that critical consciousness and address the disparities they observed in computing.

2.2 Critical Pedagogy

Ever since bell hooks elucidated the relational struggles of bringing criticality to formal classroom settings in her fundamental work *Teaching to Transgress* [28], this concept has increasingly gained traction throughout education, inspiring the rise of critical pedagogy.

Synthesizing critical pedagogy and sociocultural theory, Vossoughi and Gutiérrez called for a focus on the micro-processes of critical pedagogy, meaning *how* we teach or the practice of critical pedagogy, in addition to what we teach or the content of critical pedagogy [70]. By turning our attention to micro-processes of critical pedagogy, we may better understand how social relations are developed, how power and ideologies are embedded in practice, how tools expand or limit opportunities for the advancement of critical thought, and how learners develop as thinkers and historical actors.

For instance, in their investigation of the micro-processes of critique and dialogue in literacy classrooms, Taylor and Hikida uncovered tensions inherent to the practice of critical pedagogy within the oppressive reality of classrooms, which have to grapple with limited resources, standardized testing pressures, and other constraints [63]. Their findings exemplify the contradictions of critical theory, which aims to critique and transform society by combining normative perspectives with analysis of society's conflicts, contradictions, and tendencies [30, 35]. Heeding Vossoughi and Gutiérrez's call, our study aimed to investigate the processes of critical pedagogy within a computing classroom, and to reveal potential emergent tensions of critical computing pedagogy in practice.

2.3 Ideological Orientations around Computing

The dominant narrative around technology is that they are neutral and value-free tools that can be wielded for good or evil. This narrative manifests in the technosolutionism underlying many computing education initiatives, that technology holds the answers to all of society's challenges [54, 58]. In contrast, Winner argued that technologies are inherently value-laden and political, embedding the goals, values, and politics of their designers [75].

Scholars, such as Ruha Benjamin and Virginia Eubanks, have showcased the values embedded in the computing systems indelible in society. Benjamin highlighted how social hierarchies, especially racism, are imbued in a variety of internet-based technologies [4], while Eubanks detailed how flawed assumptions about race, class, and gender underlie high-tech development [13] and how economic discrimination is embedded in data mining, policy algorithms, and predictive risk models [14]. The idea that technology is always value-laden grounded the pedagogy used in this study and may be a foundational literacy in critical computing education.

2.4 Narrative Identity and Storytelling

Narrative identity is an individual's internalized and evolving life story, incorporating the reconstructed past and imagined future [46]. To capture the sliver of our participants' narrative identity that pertains to computing (as our participants' relationship with technology are only one part of their life story), we utilized storytelling techniques. Rooted in indigenous knowledge, storytelling techniques facilitate narrative-driven learning, enabling students to learn by connecting with their own as well as one another's stories [74] and centering values such as respect, responsibility, reverence, reciprocity, holism, interrelatedness, and synergy [3]. Storytelling is also commonly used in technology design to encourage individuals to reflect on their past to envision future experiences [5, 7]. Within computing education specifically, students, teachers, and researchers have used storytelling and speculative design to interrogate existing social inequities perpetuated by technology and imagine more liberatory futures [20, 26].

Leveraging storytelling to elicit our participants' narrative identity framed the design of the prompt given to our participants at the start of the class, where they thought of the first adjective that comes to mind when they think of technology and wrote a story of when technology was that adjective for them. For our participants' final projects, we returned to this framing, but specifically with digital storytelling, a technique to assist people in using tools of digital media to craft, record, share, and value the stories of individuals and communities [1]. We prompted our participants to tell speculative fiction stories of either alternative histories or imaginative futures grounded in their relationships with technology. By drawing from narrative identity and storytelling, we aimed to (1) elicit our participants' perceptions of and attitudes toward both the benefits and perils of technology to understand where they might be in their sociopolitical development at the start of class and (2) provide our participants opportunities to address the technology-related disparities they observed and re-imagine alternative narratives in their final projects.

3 Related Work

We first situate this study within the broader critical digital citizenship education literature, which have long wrestled with issues of equality and justice in digital contexts. We then turn our attention to related scholarship in computing education: culturally responsive computing and critical computing.

3.1 Critical Digital Citizenship Education

There have been long-standing efforts to educate adolescents on both the benefits and risks of computing to their lives, largely in the form of individual safety and civility through digital citizenship education. Digital citizenship is defined as the responsible and respectful use of technology to engage online and find reliable sources [49]. Although the focus of this article, computing, is defined more broadly as goal-oriented activities requiring, benefiting from, or associated with the creation and use of computers [9], the digital citizenship literature has historically been more focused on supporting youth to reason about issues of safety, reliability, responsibility, equality, and justice in digital spaces [18]. We therefore draw from this body of work because of this study's similar focus on supporting youth to interrogate the values implicit in computing that perpetuate societal injustices.

There is a wealth of literature on teaching adolescents digital citizenship, covering a wide range of topics such as age-appropriate pedagogy [31, 48], teacher professional development [25, 37], and the translation of research into classroom practice [16]. As a result, scholars have observed that primary and secondary teachers understood computing ethics primarily in the form of digital citizenship, proposing digital citizenship as a potential area for integrating framings of computing that interrogate its embedded values [29].

Nevertheless, in their synthesis of nearly 30 years of literature on digital citizenship, Garcia et al. argued for digital citizenship efforts to move beyond a focus on individual skills and behaviors, in favor of structural factors as they manifest in discourse and reasoning in the digital sphere [18]. In addition to individual safety and civility, they identified two other forms of digital citizenship education from the literature: (1) information analysis, which centers the reasoning skills that students need to understand and analyze the source, nature, and purpose of the information they encounter online, and (2) civic voice and engagement, which centers the reasoning and discourse skills that students need to leverage online platforms to develop and share their perspectives on civic issues and engage in authentic civic participation. Across all three forms of digital citizenship education, they advocated for further emphasis on social, political, and cultural contexts, and shared normative values. They also highlighted the need for digital citizenship education to engage more meaningfully with structural issues such as racism, misogyny, and heterosexism, as well as offer pathways for civic discourse that are not cordoned solely by considerations of safety and civility. For instance, to support students in understanding the structural contexts embedded within information and data, Philip et al. developed a framework of racial-ideological micro-contestations. This framework characterized the debates about race that arose from the integration of public media data visualizations into K-12 computing classrooms [52]. Aligned with these goals, we sought to situate our computing instruction within the cultural, political, and social realities of our students.

3.2 Culturally Responsive and Sustaining Computing

Upholding the idea that technology is inherently value-laden [75], research in culturally responsive computing have demonstrated how technology can reflect the values of marginalized communities, rather than the values of dominant communities. Scholars have leveraged a range of values and practices to combat injustices in computing. Examples include Bennett et al. developing biographical prototypes to center people with disabilities in the design process [5], Eglash et al. grounding mathematical and computing concepts in cultural context [11, 12], and Pinkard leveraging vernacular culture to scaffold computer-supported literacy learning [53].

Ladson-Billings has since revised her influential theory of culturally responsive pedagogy to culturally *sustaining* pedagogy, shifting from simply reflecting the values of marginalized communities, to strengthening them [41]. Similarly, scholars have advanced the discourse around culturally responsive computing. Scott et al. identified three weaknesses of culturally responsive computing: (1) limited attention to intersectionality, (2) focus on adults designing culturally responsive technologies *for* youth rather than *with* or *by* youth, and (3) lack of guidance for assessing culturally responsive computing [61]. They therefore recommended a more nuanced conceptualization of culturally responsive computing that advocated for intersectionality, youth-driven technical innovation, and technosocial activism. A study by Garcia et al. on girls of color revealed that attention to intersectionality in culturally responsive computing positively influenced girls' perceived ability to change their educational conditions and led to a better understanding of how race and gender have influenced their personal academic goals [19].

Eglash et al. critiqued the superficial appearance of culture that may arise from culturally responsive computing and proposed counter-hegemonic computing [10]. Counter-hegemonic computing aims to meaningfully integrate counter-hegemonic practices—forms of value that low-income, African-American, Native American, and Latinx communities possess—into computing. They defined three principles for counter-hegemonic computing: (1) having both individual and social levels of analysis, (2) including not just what biases to avoid, but what computing should actively promote, and (3) using computing to advance the value of counter-hegemonic practices.

3.3 Critical Framings of Computing

Similarly driven by making explicit the values embedded in technology, critical framings of computing emphasize how technology can reflect oppressive values and structures, interrogating the "unquestionable good" of technology underpinning most of computing education. Vakil advocated for the interrogation of the sociopolitical context of the content of computing curricula, the design of learning environments, and computing education reform [65]. Similarly, Kafai et al. argued for a critical examination of the skills and competencies that make up "computational thinking" [33] and Yadav et al. proposed that computing education focus on communities, criticality, and citizenship to truly center justice [76].

Scholars have begun investigating applications of critical framings of computing in practice. Johnson et al. investigated the outcomes of an adult data literacy workshop through the lens of critical race theory, leading them to re-evaluate the data literacy ideas foundational to the workshop [32]. Similarly, Kirdani-Ryan et al. integrated critical counternarratives into a university computer systems course, finding that almost all their students expressed awareness of structural problems within computing [36]. At the secondary level (ages 14–18), Ryoo et al. detailed the practice of learning and teaching the social impacts and ethics of computing, finding that students' perspectives and agency must be centered [57]. Everson et al. similarly highlighted the significance of student perspectives and agency in their class, finding that they had to navigate students' safety and positionality before engaging in critical computing pedagogy [15]. Lastly, Mayhew and Patitsas identified three approaches to critical computing education in existing practice: developing students' critical consciousness to see structures of oppression, helping students in learning technology that supports their activism, and changing conceptions of computing practice by incorporating sociopolitical forces [45].

Similar to prior culturally responsive and critical computing efforts, we also centered our participants' prior knowledge, dispositions, and agency in our pedagogical choices, such as allowing participants to choose the topics covered in the course and providing opportunities for participants to express their interests and perspectives in their classwork. This study aims to build upon prior work by investigating how the critical pedagogy methods for secondary education proposed in *Critically Conscious Computing* [38] would manifest in classroom practice and what pedagogical content knowledge may be necessary for critical computing education.

4 Methods

4.1 *Critically Conscious Computing* as Our Instructional Basis

We based the instruction used in this study on *Critically Conscious Computing* [38], a textbook of methods for teaching critically conscious computing at the secondary level (ages 14–18). As part of a broader methodological and pedagogical movement in computing education away from an idealistic vision of computer science, this textbook aimed to make explicit the values embedded in computing that have reified oppressive structures and to rethink the underlying values of computing education to advance a more just vision for computing. Rather than delineating specific skills and competencies for critical computing, this textbook operationalizes "critical computing" as an epistemological stance toward computing, where technology is always value-laden and those values have societal implications [75]. Through this epistemological stance, this textbook reframes common topics in secondary computing education and proposes methods to teach these topics that surfaces and interrogates the values implicit in those topics.

4.2 Study Context

This study occurred during a 6-week summer program (June–August 2022) at a northwest United States university targeted at students ages 14–18 from local under-resourced schools. Students were from low-income backgrounds and/or would be the first in their families to pursue a post-secondary education. They received official school credit for courses they took in the summer program. However, this program differed from the academic school year in several ways. First, teachers nominated students for the program. Second, students received lunch allowances and a stipend for the opportunity cost of a summer job. Third, it aimed to develop community among students through study groups and field trips. Fourth, all course materials needed to be browser-based because students had different school-administered computers and university lab computers wiped locally saved data every 24 hours. Lastly, leaders of the summer program discouraged homework for elective courses to allow students to enjoy the summer break.

The first and second authors were the lead instructor and co-instructor respectively of a computing-based elective course in the program. Elective courses had one hour of class time four times a week across all six weeks. The university institutional review board granted this study exemption because this study was part of instruction, which meant that we did not need to obtain parental consent. However, we maintained informed assent by enabling students to opt out or assent to various levels of participation after detailing the nature of the research.

4.3 Participant Demographics

After we described the risks of research participation (namely discomfort from discussing potentially negative experiences with computing), 15 of the 20 students in the course assented to their classwork being analyzed for research through a form administered on Day 1 of class. The form also included free-response questions for participants to self-describe their gender identity, ethnic identity, languages spoken at home, disabilities, and other aspects of their identity they would like the instructors to know. We also asked participants who assented to choose a pseudonym; if they did not provide a pseudonym, we used an initial. The age range of participants was 15–17 years old. To preserve their anonymity, one participant described their disability as fatigue and nausea and one student identified as Muslim. The rest of the demographic information is shown in Table 1.

We intentionally did not ask for any information about their prior computing education experiences because previous work has associated perceptions of having less prior experience with a

Table 1. Participant Demographics

Pseudonym	**Gender**	**Ethnicity**	**Languages Spoken at Home**
A	Male	Hispanic/Latino	English and Spanish
Batman	Woman/Female	Black, Somali	English and Somali
Becky	Female	Mexican	Spanish
Bob	Female	Vietnamese	Vietnamese and English
Dairy	Male	Black/Somalia	English/Somalia
J	Female	Half Korean, Half Argentinian	English and Korean
Jake	Male	Somali	English/Somali
Kevin	Female	Vietnamese	Vietnamese
Miranda	Female	Somali	English
R	Girl (she/her)	Asian	Vietnamese
Sophia	Girl (she/her)	Chinese American	English and Chinese (Toisanese/Taishanese)
Stewart	Male	Laos	Laos
T	He/him	Asian	Vietnamese
Z	Male	Afghan	Pashto, Dari, English
Zoro	Male	Vietnamese, Asian	English and Vietnamese

lower sense of belonging and confidence, especially for students who identify as women or underrepresented minorities [44, 60]. As nearly all our students had either or both of those identities, asking them about their prior computing education experiences may have negatively influenced their participation in both the study and the course.

4.4 Study Timeline

In the month leading up to the start of the summer program, the first author prepared to teach the course by (1) reading *Critically Conscious Computing* [38], and (2) planning the adaptation of the unit sketches in the book chapters for the summer program's structure (four 1-hour classes per week) with the feedback of other computing education researchers, including the fourth and fifth authors who had both previously taught in the summer program. Two weeks prior to the start of the summer program, the first author trained the course's two co-instructors, including the second author, by having them review the textbook and the plan for adapting the unit sketches, and then discussing any questions or feedback. As a result of the source material, the instructors approached teaching computing content with a critical stance. However, this did not mean that we performed direct instruction of critical computing ideas; instead we facilitated students in developing their own critiques of technology as part of learning computing.

The form given on Day 1 of class, in addition to asking about research assent and demographic information, prompted participants to select their top three topics of interest adapted from the chapters in the *Critically Conscious Computing* textbook [38]. We described the topics in age-appropriate language for participants with no prior computing knowledge.

On Day 2 of class, our goal was to prompt participants to reflect on their experiences with computing technologies (RQ1). We started with a class brainstorm of what was included in computing technologies to encourage participant to think beyond more "apparent" computing technologies, such as their laptops and smartphones. After the brainstorm, we set expectations that the reflection assignments would receive full credit if submitted, so they could write freely without the pressure of a grade and "perfect" English grammar and spelling (good pedagogy in general, but particularly important as participants had varying English fluency). Participants then wrote reflections to the two prompts below. Given our participants' varying English fluency, we provided examples of adjectives unrelated to computers in the second prompt, so as to avoid priming participants to choose adjectives similar to the examples. After writing a reflection for each prompt for 15 minutes, each instructor facilitated a 10-minute discussion with a small group consisting of 6–7 participants.

Day 2 Reflection Prompts.

(1) Describe a day in your life and how you interact with computing technologies.
  (a) What computing technologies do you use the most?
  (b) What role does computing technology play in your life?
  (c) How do you feel about the role computing technology plays in your life?
(2) What's the first adjective that comes to your mind about computers?
  Adjectives are words that describe something, such as "fluffy," "smelly," and "delicious."
  (a) Why does that adjective represent computers for you?
  (b) Tell me a story about when computers were <that adjective> for you.

After covering algorithmic bias in Days 3–6 of class (the only instructor-selected topic; more details on the methodology and findings of the scaffolded sensemaking activities in [59]), we covered the three most popular topics from [38] indicated in the Day 1 form responses: *Programming Languages (PLs)*, *Programming*, and *Projects*. Classes typically adhered to the following structure. At the start of class, instructors introduced participants to the relevant concepts, modeled the activity that participants would engage in, and discussed both the process and potential outcomes of such activities with participants. Next, participants would complete these activities either independently or in pairs, guided by worksheet prompts adapted from the unit sketches in [38]. At the end of class, instructors would bring participants back together to discuss their experiences and outcomes from the activities, either as a class or in small groups. Table 2 summarizes the course timeline and the adaptation of proposed class activities from [38].

Throughout the course, we took measures to mitigate the power dynamics between the instructors and participants. First, we emphasized that all participants would receive full credit for their assignments if submitted. This reduced the likelihood that participants would write what they perceived instructors wanted, and instead encouraged them to write their authentic perspectives. Second, as the course integrated reflections and discussions of oppressive power structures in computing, the instructors offered their own experiences navigating such power structures, both proactively and in response to participant questions.

On the final day of instruction (Day 23), we prompted participants to write reflections on their favorite and least favorite part of both working on their final projects and the course as a whole, followed by the second prompt from the Day 2 Reflection Prompts (above). The rest of class time was spent making revisions to their final project in preparation for demos to the whole summer program, not only the students enrolled in this computing elective.

4.5 Data Collection and Analysis

Our primary data source was participants' classwork, because most students (15/20) assented to their classwork being used for research, as compared to only 7 students who assented to group discussions being recorded for research. For the *Computing Experiences* reflections and the *PLs* unit, we collected classwork from 15 participants. For the *Programming* unit, *Final Projects*, and the *End of Course* reflections, we only collected work from 14 participants, because R left the summer program.

Table 2. Course Timeline and Adaptation of Proposed Class Activities from *Critically Conscious Computing* [38]

Day	Unit(s)	Class Activities
1	Course Introduction	Form for research assent, demographic information, and class topic selection
2	Perceptions of and Attitudes toward Computing	Computing Experiences Reflections
3–6	Algorithmic Bias	Scaffolded Sensemaking Activities [59]
	PLs	
6	Origins of PLs	Research on the invention of Python and Python community of practice
7–8	How Smart are PLs	Modification of example code in Python or Javascript to critique output and error messages
9	Comparing PLs	Design critique of Python, Javascript, Snap!, and C++.
	Programming	
10	Programming Introduction	Critique and refinement of example code for financial aid calculation
11	Programming Implementation	Implementation of students' refinements to the example code
12	Programming Testing and Debugging	Iteration on their code to find and fix bugs
13	Programming Reflection	Reflection on the programming process and critique of their code
	Projects	
14–15	Final Project Design	Brainstorm, selection, and scripting/storyboarding of digital speculative fiction stories on alternative histories or imaginative futures.
16–17	Final Project Requirements and Planning	Definition of project requirements and milestones
18–21	Final Project Implementation	Implementation of projects in students' choice of PL and platform
22	Final Project Demos	Class-wide demo of projects
23	Perceptions of and Attitudes toward Computing	End of Course Reflections and Final Project Revisions for Program-wide Demo

We qualitatively coded participants' classwork as our data source, using a combination of inductive and deductive coding. Instead of capturing statistical agreement metrics such as inter-rater reliability [47], we resolved disagreements by building consensus through discussion, following Hammer and Berland's position on qualitative coding that positions qualitative codes as tabulations of thematic claims about data [24]. Consistent with this position, we do not treat qualitative codes as data and therefore, we only report the number of participants who mentioned a specific code (presence/absence), but not the number of times each participant mentioned a specific code across our actual data (their classwork).

To develop our coding manual, the first four authors inductively open-coded a subset of participant classwork to identify potential codes. Next, the first four authors collaboratively identified patterns in the codes using affinity diagramming and then discussed definitions and inclusion/exclusion criteria for each code, culminating in the first draft of our coding manual. The first four authors then tested this manual by each coding different subsets of participant classwork. After testing the coding manual, we refined the definitions and inclusion/exclusion criteria, resulting in

Table 3. Coding Manual

Code	Explanation
Computing Reflections (Start and End of Class)	
Ease of Technology	The ease technology brings to their lives and/or how easy technology is to use.
Access to Technology	The privilege of having access to technology around them.
Capabilities of Technology	What technology can do, such as computational power or the future capabilities of technology.
Perils of Technology	The negative effects of technology on their lives, communities, and society.
Self-efficacy	Confidence in their abilities and/or work.
Student Agency	Students' own agency in the class, computing, or in their relationship with technology more broadly.
Human Agency	Human's role in shaping technology, acknowledging the humans behind the creation of technology.
Other	Code that was not covered by the previous codes.
Description	Description of the "other" code.
Topic Assignments	
Each of Technology	Same as in Computing Reflections
Perils of Technology	Same as in Computing Reflections
Limitations of Technology	Limitations, shortcomings, or mistakes made by technology.
Assumptions about Technology	An assumption about a technology based on its design, e.g., if a technology looks abstract, it must be more advanced.
Spectrum of Trust between People and Technology	Trust in humans, trust in technology, or somewhere in between both ends of the spectrum, e.g., belief that technology can solve everything, or that humans should have decision-making power.
Belonging	Contributors to a sense of belonging.
Self-efficacy	Same as in Computing Reflections
Censorship	Descriptions of ways to suppress information, e.g., book banning.
Social Consciousness	A broad awareness of social issues, e.g., representation, marginalization.
Characteristics of People Impacted by Technology	Descriptions of characteristics of people impacted by technology, e.g., age, disability.
Difficulties addressing Harm	Difficulties with identifying, addressing, or resolving harm or bias in technology.
Other	Same as in Computing Reflections
Description	Same as in Computing Reflections

We analyzed for any mention of the codes above, including negative mentions, such as the lack of access, in addition to access to technology.

our final coding manual (Table 3). With the determined codes in the manual, we then deductively coded all participant classwork.

In qualitatively coding classwork, we adopted the principle of coding responses "as-is," meaning that a phrase in isolation is enough to justify coding it a certain way, to limit inference by the first four authors and to center participants' voice in the classwork. We formed coding pairs consisting of one author who taught the course (first and second authors) and another who did not teach the course (third and fourth authors) to have different perspectives in the analysis, especially because the authors who taught the course were familiar with the participants. Each pair coded classwork from distinct subsets of participants, so that at least two authors analyzed all classwork and documented their disagreements. We resolved disagreements through discussion, first within the coding pairs and then if consensus was not reached or additional input was required, disagreements were discussed with all four authors. Codes that required the most discussion were:

(1) "*assumptions of technology*" because the subject of the assumption was sometimes ambiguous,
(2) "*characteristics of people impacted by technology*" because it was sometimes unclear if the cause of the impact was indeed technology,
(3) the distinction between "*social consciousness*" and "*characteristics of people impacted by technology*" because the former describes a broad awareness of societal issues while the latter includes detailed descriptions of people impacted by technology, and
(4) "*belonging*" because participants were prompted to discuss a potential sense of belonging, simply mentioning a sense of belonging, or lack thereof, was not sufficient to be included. We needed to identify participants' *reasons* for their potential sense of belonging, which was sometimes obscure.

After all classwork was coded, the first four authors collaboratively conducted a post-hoc analysis to synthesize the relationships between the codes. In the post-hoc analysis, we noted at which points the codes emerged in the course and how the codes related to one another (e.g., did some codes tend to emerge from the same assignment? if so, how?). These relationships between the codes are further detailed in the results.

4.6 Positionality

Positionality statements allow for clarity in the relationships between identities of the authors and both the research topic and the identities of the participants [27, 43, 55]. Each author wrote their respective statements to delineate experiences and perspectives that influenced their research engagement.

Similar to many of the students, the first author is an immigrant, grew up with economic challenges, and was the first in her family to pursue a university education in the US. She also shared a racial identity with half the students. In her decade in computing, this was the first classroom she had been in consisting entirely of students of color and felt an immense responsibility as the lead instructor of the course. She collectively had 10+ years of experience teaching primary students (ages 7–14) informally and undergraduate students (ages 18–22) formally, but this was her first time teaching secondary students (ages 14–18). She relied on the fourth and fifth authors for guidance, because they both had experience teaching secondary students. Her identity and lived experiences with systemic marginalization in computing and society led to her interests in critical computing literacies for youth. Her assumptions around critical computing literacies stemmed from her educational experiences and research training in computing cultures that championed utopian visions of computing and prioritized technical prowess above all else. Throughout the entire research project—from design to analysis—she actively and reflexively worked to deconstruct, interrogate, and unlearn the values implicit in the computing cultures she was trained in. She was the leader of this research project.

The second author had 3 years of experience teaching computer science to both secondary and undergraduate students and received guidance from the other authors who had more experience with co-instruction and critical pedagogies. She is a white computer science student whose experiences learning primarily from direct instruction and witnessing an abstraction of computing from its impact motivated her to join this project to investigate how to improve computing curriculum and encourage more reflection, agency, and sense of belonging for all students in computing.

The third author had 6 years of experience teaching computer science in after-school and summer camp environments. Their personality as a Black immigrant graduate student inspires the work they do to broaden participation in computing. They participated in this project because they were interested in analyzing how a class of underrepresented students developed critical consciousness.

The fourth author had 10+ years of high-school teaching experience and embraces co-constructed pedagogy because of the importance of student voice in learning, and the organic ways that co-construction allows instructors to interrogate their power in educational settings. She is a white graduate student who joined this project because she was curious about the application of the book [38] into the classroom.

The fifth author, a multiracial researcher and CS educator, brought 10 years of experience working with the study participants' communities, as well as a curiosity about how the teaching methods explored may or may not resonate with them. Her biases in the work stemmed from her career-long experiences in direct instruction, her nascent experience with more co-constructed pedagogies, and her frustrations with the role of technology in perpetuating racial hierarchies and binary systems of gender. She viewed this project as a way to further deconstruct her bias toward direct instruction, exploring more student-centered, critical pedagogies, and a way to learn how students might gain critical literacies about computing. She brought to the work a view of critical pedagogy as one that begins in students' lived experiences, broadens with awareness, and deepens with action. She also brought her own struggles with balancing criticality with optimism in the classroom, and managing the care-giving and boundary setting work inherent to examining student positionality.

5 Results

Overall, our observations revealed that our participants brought a range of existing narratives about computing often grounded in their identities and lived experiences, from enchantment with technology's capabilities to reticence around technology's ubiquity. We discuss these in detail, starting with their initial perceptions, then describing how each of the units revealed different tensions that emerged when learning computer science topics through a critical lens. We refer to our participants' quotes using their chosen pseudonyms (Table 1).

5.1 Computing Perceptions and Attitudes at the Start of the Course

At the start of the course, we asked our participants to reflect on their initial perceptions of computing by describing a day in their lives and how they interact with computing technologies. They also asked what was the first adjective that came to mind when thinking about computers, why that adjective represents computers to them, and to tell a story about when computers were <that adjective> (see Section 4 for specific prompts).

Through our data analysis of our participants' start of course reflections, we observed several themes of our participants' initial computing perceptions and attitudes which featured a spectrum ranging from the benefits to perils of technology. Our participants mostly expressed that technology was a beneficial because it was easy to use and had a lot of capabilities, reifying prior work on adolescents' mostly positive relationships with technology [2, 56, 69]. However, some also expressed that they experienced negative impacts on their lives through compulsive technology use and over-reliance on technology. Additionally, several participants also perceived access to technology as a privilege.

The benefits of technology centered around capabilities of tech, ease of technology, and access to technology. Many participants (13/15) wrote about the capabilities of technology and the computational power of computers. Batman said,

> "*Everything about computers is so complex even at times where you think it isn't. The way that there are dozens of languages we can communicate to computers with is so intriguing to me. I constantly wonder about the significance of it all and why it matters.*"

Similarly, Stewart touched on the capabilities of computers to perform complex tasks and also the abstraction of technology. Stewart described computers as robotic because:

> "*it feels like there is a robot inside of the computer doing the work for me.*"

Many of our participants (11/15) also discussed how technology makes their lives easier and is easy to use. Z explained,

> "*We as humans are trying to make are life easier and easier each day and a way we do that is by using computing technologies.*"

Another participant perception of technology included access as a privilege (7/15). Becky explained,

> "*I feel good because we are privileged that we have technologies around us. Makes lots of things easier including HW, School, and our everyday life.*"

However, our participants also perceived access to technology as a peril (7/15), since it is not always positive or guaranteed due to costs. Stewart explained,

> "*[Computing technology] plays a big role in my life because I use the internet a lot and so does my family. I feel good and guilty because sometimes I feel like I'm having fun while using the internet, but other times I think about how my family has to pay for the internet and how I use so much.*"

Our participants' initial perceptions on the perils of technology additionally focused on compulsive technology use (5/15) and over-reliance on technology (7/15). While ease of use was seen as a positive aspect of technology for some of our participants, others commented on the over-reliance of technology. Dairy explained:

> "*I feel that it brings ease and accessibility to the world, but it is also scary how we only rely on it for our everyday lives. I wonder what would happen if our technology just stopped for one day.*"

Zoro explained the compulsive technology use he had witnessed:

> "*I honestly think that those two adjectives [attached, obsessed] match technology the best... Everyone nowadays has some sort of technology on them, it's either a phone, smart watch, laptop, or airpods. People are too accustomed to technology that its like a drug.*"

This diversity of sentiments revealed that our participants brought to the course numerous narratives about technology, from its utility, unequal access, and risk of addiction. These narratives also suggest that participants might have started the course at the acritical or adaptive stages of sociopolitical development around technology, which could shape their engagement with the critical computing ideas presented in the course.

5.2 Tension between Social Consciousness and Computing Knowledge

We observed a tension between our participants' social consciousness and computing knowledge as they were introduced to the design of PLs. The PL unit consisted of three assignments. In the first assignment, our participants explored the origins of PLs by researching the inventor of Python and the Python community of practice. In the next assignment, our participants assessed how smart a

Fig. 1. Example code used in the PLs unit in Python (top left), Javascript (top right), C++ (bottom left), and Snap! (bottom right).

PL was through manipulating example code and reporting what errors ensued. Our participants could choose between JavaScript or Python. In the final assignment, participants compared four PLs—Python, JavaScript, Snap!, and C++ (Figure 1). We chose Snap! instead of the popular block-based programming language Scratch because we wanted the PLs to have as analogous capabilities as possible. For instance, higher-order functions can be written in Snap!, but not in Scratch.

In these assignments, we observed that our participants conceptualized the social contexts surrounding PLs. We found that our participants identified assumptions that PL designers made about the users of a PL, such as prior knowledge, physical abilities, and language fluency. After inviting them to explore Python's community of practice, our participants also expressed their assumptions about a PL's user community, which seemed to affect their potential sense of belonging within that community. By communicating these assumptions, our participants seemed to begin to identify the underlying values embedded in the design of these PLs, reflective of the epistemological stance of critical computing. Nonetheless, participants did not go far beyond a simple identification of these assumptions, perhaps because they did not necessarily have the computing knowledge to further interrogate these assumptions and values.

For example, one prevalent sentiment was that PL designers made assumptions about their intended audience (11/14). Our participants saw the potential barriers users of various PLs may face when programming. In the third assignment, Bob mentioned that she thinks Python, JavaScript, Snap!, and C++ would be:

> "*harder for people from other countries that don't speak English to learn.*"

Additionally, while exploring Snap!, Kevin wrote,

> "*it's a different format than traditional code, simple language can cause the outcome to be inaccurate, color coding means that it might be difficult for colorblind people to use.*"

The PL unit responses also revealed how our participants perceived audience through its capabilities and communication. Batman compared the intended audiences of Snap! and C++, stating that:

> *Snap! is colored and block based, making it the best option for absolute beginners in CS. [...] C++ is suited for someone who is already familiar with CS and feels confident in their ability to understand code.*

In a similar vein, Jake remarked,

> "*I think this programming language is pretty smart, not only is it able to stop errors in your code but it can pinpoint exactly what is wrong most of the time. One thing I don't understand is why they use weird vocabulary when describing the error, wouldn't it be easier and faster to understand if they just explained it normally?*"

Some of our participants surfaced assumptions about the communities of a PL (5/14). In the first assignment, when asked if they felt as though they belonged in the Python community some participants wrote that they felt as though they belonged, while others did not. R mentioned,

> "*I think the Python community is a space where I belong because it has detailed information that makes it easy for me to learn and I can visualize how to code through example images.*"

The organization of informational material increased R's sense of belonging. In contrast, Sophia wrote,

> "*I also kind of see the community as unwelcoming, not many women of color are in this field (though there are more and more everyday!). This community is most likely well connected and that would mean access to help would be unlimited and would reach a lot of people.*"

Here Sophia notes impressions of the community's status quo and how it is changing, impacting her own sense of belonging.

5.3 Tensions between Social Consciousness, Computing Knowledge, Self-Efficacy, and Attitudes toward Technology

In addition to a tension between our participants' social consciousness and computing knowledge, we discovered tensions between our participants' social consciousness and their self-efficacy, and attitudes toward technology when they learned to program. The programming unit had four assignments. In the first assignment, we first introduced example code written in Python, JavaScript, C++, and Snap! that calculated how much financial aid students should receive for post-secondary education, which only accounted for state residency.[1] After critiquing the example code, our participants wrote pseudo-code to improve its design. In the second assignment, they implemented

[1]In the US, students have to pay for post-secondary education, with governments and educational institutions offering different levels of financial help depending on the student. Students who pursue post-secondary education at institutions funded by their state often pay less.

```
if isInState:
  financialAid = financialAid + 50

if householdIncome < 100000:
  financialAid = financialAid + 25
if disability:
    financialAid = financialAid + 5
if studyAbroad:
  financialAid = financialAid + 30
if scholarship:
    financialAid = financialAid - 20
    Tuition = ["Disability- 5% off Tuition", "Study Abroad- 30% off Tuition", "Scholarship- 20% off Financial Aid"]
    for x in Tuition:
      print(x)

print("This student is getting "+ str(financialAid) + "% off tuition")
```

Fig. 2. Subversive financial aid code written by A & T. It prints deceiving output that students with disabilities, study abroad, and scholarships get help with tuition. However, if students receive help with tuition for any reason ("scholarship" variable), the school charges them an extra 20%.

their code and in the third assignment, they tested it. After these activities, they completed a reflection for the fourth assignment.

In this unit, we found that even with limited program reading experience, our participants identified various ways that the example code could lead to exclusion or harm. However, when asked to write their own code to resolve those issues, some of our participants communicated challenges in addressing all the issues they identified, while others did not identify issues in their own code.

This was partly evident in how nearly all of our participants (13/14) explicitly grappled with the impacts of technology on humans and society, reflecting their social consciousness. For example, Stewart expressed that the example code for computing financial aid was:

> "*biased because it doesn't consider how many people are in the household.*"

Similarly, Z wrote that:

> "*If the household makes less than $40,000 then they should get 15% more discount. These factors should be applied to both in state and out of state.*"

Some of our participants (8/14) shared that they were thinking about different people impacted by technology. For example, A & T worked together for this unit. In the programming introduction, they remarked that they

> "*should add a disability so as to make it easier for students who have a disability. We believe that it should take another 10% off.*"

A & T later cheekily flipped the script when completing the assignment, demonstrating an understanding the subtle effects of financial aid algorithms (Figure 2). A shared,

> "*we decided to make a scam school causing there to be a huge injustice for students that aren't out of state, study abroad, have a disability and have a scholarship.*"

Similarly, T reflected,

> "*It was easy to do harm while I was programming when I can put the little - [minus] sign instead of + [plus] sign so students that have scholarship will pay more than others.*"

This subversion of the assignment seemingly contradicted our assumptions about what critical computing ought to be, with A & T writing code that amplified hegemonic values and deliberately caused harm. However, their ideas suggested expansions for what critical computing instruction can include, that perhaps having students try embedding both hegemonic and counter-hegemonic values into technology may offer opportunities for deeper interrogations of the values implicit in the design of technology.

In the third unit, some of our participants showed signs of an emerging computing self-efficacy (13/14) but expressed difficulties addressing harmful values in and with code (8/14). This revealed a tension between their social consciousness, and their computing knowledge and self-efficacy as they moved from simply identifying the potentially harmful values reflected in code to addressing them. In their programming reflections, many of our participants, like Kevin, Zoro, and Z reflected that their favorite part of the unit was coding. Z remarked that they:

> "*liked the part where we were able to create our own code and express our thinking using coding.*"

Although our participants learned to code and could identify how code could perpetuate exclusionary or harmful values, some described difficulties in addressing those issues with code. These difficulties could have stemmed from an actual gap in computing knowledge or a *perceived* gap, which would reflect a lack of computing self-efficacy. For example, J wrote:

> "*I think that the people that are above 500,000 shouldn't have any financial aid. I didn't know how to put that in the code, so if they were in state and they get 40% but if they get over 500,000 then they get 0%*"

The third assignment surfaced the spectrum of trust between people and technology (6/14), reflecting their attitudes toward technology. Some of our participants identified roles that people and tech should play in determining financial aid. For example, Miranda and Becky wrote that the computer should:

> "*Collect the data, calculate the percentages, the calculation should probably be up to the computer*"

However, they also noted that people should make some of the decisions:

> "*I think the part where they assess the household income and household residents should be up to the humans to make.*"

Although our participants identified potential harmful values in the code they were given, they rarely identified harmful or exclusionary values within their own code. For instance, in this unit, Z chose to write code to decide how much financial aid to give new businesses, rather than post-secondary students. In his reflection, Z wrote,

> "*I don't think it will cause any injustice because we tried not to be biased or give too much to another business in our code.*"

Similarly when asked what potential harmful values from the original design persisted in their implementation and how they could have been mitigated, Dairy and Jake simply wrote, "*None.*" Such responses may be indicative of their self-efficacy, that they felt confident that they could write code that would not do harm, or their attitudes toward technology, that after they resolved the possible harmful values embedded in the code, the code was now "harmless."

5.4 From Awareness to Action

In the final project, we wanted our participants to have an opportunity to express their relationships with and narratives of technology. Our participants used digital storytelling to create a speculative fiction story about technology either as an alternative history or imaginative future. They worked with a partner of their choosing. Over the last 2 weeks of the course, pairs checked in with an

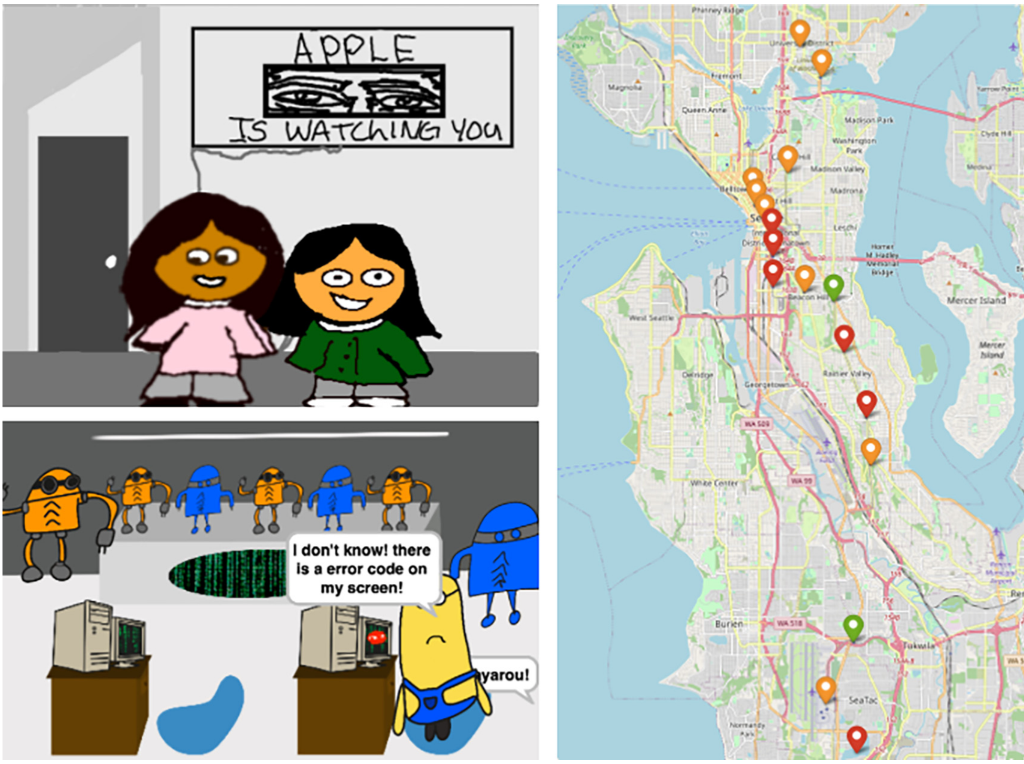

Fig. 3. Becky and Miranda's project on surveillance (top left), Kevin and Stewart's project on programming robots and other future technology (bottom left), A & T's teleportation project (right).

instructor while planning and implementing their story using a variety of tools and PLs. Throughout the project, we prompted them to reflect on who they represented in their story, and who they omitted, and how people with different knowledge and experiences might engage with their project. At the end of the course, our participants presented their projects to each other as well as other students in the broader academic program.

Our observations of their projects indicated that our participants had a wide variety of relationships with the narratives they expressed through code. These ranged from narratives on wariness about the dangers of technology to narratives with an idealistic technosolutionism, reflecting various ideological orientations toward computing.

An analysis of our participant project designs and planning surfaced their perceptions of the ease and convenience brought by technology (6/14) as well as the perils of technology (4/14). Additionally, many projects focused on social consciousness (8/14) and several touched on the characterizations of the people impacted by technology (4/14). For example, Becky and Miranda's project focused on the negative impacts of technology, especially surveillance (Figure 3). They created an animated story using Snap! and described their project as:

> "*Two roommates working for a big tech corporation*
> *Big tech corporation spies on its civilians which is everyone in the world*
> *Two roommates seem to be the only ones conscious and independent from a technology consumed life*

TVs and advertisements on every corner
Roomates run away not before getting in trouble with the corporation."

Through this story, Becky and Miranda indicated an awareness of social wrongs from technology (e.g., surveillance), as well as some ideas of how to potentially address that wrong, with the two roommate characters resisting a "*technology consumed life.*"

Other projects contained a mix of both positive attributes of technology (capabilities and ease of use) and negative attributes (perils of technology and access). A & T used Python and HTML to create an interactive map of teleportation locations to complement their story of Vladdy Patty Jr. who transformed the teleportation device his father (Vladdy Patty) created, so more people could use it (Figure 3):

"*In Vladdy's original design, it depended a lot on coordinates. Making it hard for common people to use it properly. The reason it was like this was because it was made for the military. Yet when Vladdy Patty jr. would take it to renovate it. He would work with the Moscow city government to allow certain spaces to be teleportation stations (kind a like metro buses). This would make it faster and easier for common people to use and benefit from.*"

Through their story, A & T acknowledged human agency in improving technology and how it can be used to better help people. The accessibility issues of the original design reflect their social consciousness and how technology is not always designed for everyone. In contrast, their new design allows greater ease of use and more technological capabilities and demonstrates their agency to create better technology. A & T also highlighted how collaboration enhanced the value of the teleportation device, recognizing the social context of technology's development. Through their new design, A & T indicated that not only were they aware of potential social wrongs that could come from technology, but also that they were developing knowledge of how to address those social wrongs.

Some participants chose to focus on the positive aspects of technology and illustrated their new computing skills. Stewart and Kevin created an animation using Snap! as follows (Figure 3):

"*Kevin makes robots in the cloud factory (in the sky).*
His coworker Stuart codes the robots with Kevin.
Kevin is excited to start coding.
Stuart finds a bug in/breaks the code and the robots start malfunctioning.
Robots go crazy and take over the world...
Kevin goes into his flying car and defeats the robots by using the gadgets.
Back in the office Stuarts rushes and deletes the code of the robots.
Kevin is promoted and he makes changes to the office.
He teaches his other minion brothers how to code and test the codes they make before releasing it to the public."

In Stewart and Kevin's project, the heroes save the day by learning to code, suggesting a technosolutionistic orientation toward computing. The characters express their excitement to code and reflect what the participants have learned in the course: how to find bugs in their own code, write their own code, and the importance of testing code. The moral of this digital story reflected the participants' positive perception that learning to code can help make the world a better place. Taken together, these participants' projects illuminate how their social consciousness, newly acquired computing skills, and attitudes toward technology can be intertwined in their narratives of computing.

5.5 Computing Perceptions and Attitudes at the End of the Course

At the conclusion of the course, we asked our participants to reflect on their computing perceptions and attitudes by explaining why they choose the story they implemented for their project, their favorite and least favorite parts of both the project and the course as a whole. Our participants again shared the first adjective that comes to their mind about computers and why that adjective represents computers to them.

We observed that despite substantial instruction in and discussion of critical narratives of computing, our participants seemed mostly firm in their positive attitudes toward technology, largely rooted in the capabilities of technology. The main differences we found at the end of the course were mostly indications of computing self-efficacy and both their own agency and human agency more broadly in the creation of technology. While they seemed to see themselves as active creators, rather than passive consumers, within their computing narratives, these narratives remained relatively stable by the end of the course.

As reflected in their initial perceptions of computing, many of our participants (7/14) continued to comment on the capabilities of technology at the end of the course. Stewart explained,

> "*I feel like computers can do anything that you need. For example, it can do math, or it can do work like I am doing right now.*"

Furthermore, several of our participants also acknowledged technological capabilities with a greater understanding of human agency and the role of people in shaping technology. Bob said,

> "*The word smart described computers for me because I find computers one of the smartest things. If the person programming it is smart, then the computer can become just as smart as them in a way. If it can take what it learned from their programmer and do more with that coding in a way, then it's smart to me.*"

At the end of the course, some of our participants conceptualized the role of programmers as inherent in the intelligence of a computer, touching on both the role of people in creating technology and the effects of technology on people.

Many of our participants (10/14) also demonstrated a growing self-efficacy with computing by the end of the course, expressing their confidence with coding projects. Jake said his favorite part of the course was:

> "*looking at our website after writing a new code and having the satisfaction of it working.*"

Our participants also emphasized collaboration as a useful tool to improve their understanding and enhance their learning process. For example, Kevin mentioned how learning from different people furthered her understanding of the material. She wrote in her reflection,

> "*My favorite parts of this course was coding the final project and discussing the problems in codes because I got to work in code and hear from different perspectives.*"

Our participants also reflected on their own agency in the course, computing, and technology more broadly. Z explained,

> "*My favorite part of the course was doing the coding project because it was great that we got to use our own imagination and thinking and implemented them into code.*"

Through the project, our participants were able to showcase the skills they had developed and improved upon in the course and expressed agency to translate their ideas into code. Our participants indicated some sociopolitical development through their expressions of social consciousness around technology throughout the course, with some participants moving beyond awareness of possible social wrongs from technology to developing knowledge of how to address these wrongs. Nevertheless, they also expressed their own narratives of the spectrum of positive and negative impacts of technology, human agency in computing, and their own growing self-efficacy.

6 Discussion

We now return to our overarching research questions: RQ1, which sought to understand the ideological orientations toward computing that may shape students' learning of critical computing ideas, and RQ2, which aimed to investigate the processes and practice of critical pedagogy in a computing classroom.

6.1 RQ1: What Perceptions of and Attitudes toward Computing Might Students from Low-Income and First-Generation Backgrounds Bring into Critical Computing Classrooms?

Our participants exhibited mostly positive perceptions of computing, largely stemming from the capabilities of technology, ease of/from technology, and access to technology. Our participants drew from their experiences with the current capabilities and their projections of future capabilities of technology. The ease of technology adoption, the convenience that technology has brought to their lives, and the availability of technology also seemed to shape their positive attitudes toward computing. This finding reified prior work that has found that adolescents tend to have more positive perceptions of technology, underestimating its negative impacts on themselves [2, 56, 69].

However, some of our participants also demonstrated an awareness of computing's negative impacts on their lives and society. While access to technology informed most of our participants' positive dispositions toward computing, some participants recognized disparities in technology access, such as the high cost of internet access at home. Our participants also expressed concerns around a perceived over-dependence of technology in both their own lives and society, while others mentioned worry over their own compulsive technology use. The varying levels of awareness of computing's adverse implications could suggest that our participants may have entered the course at either the acritical or adaptive stages of sociopolitical development, possibly shaping their learning of critical computing ideas. By understanding the potential perceptions of and attitudes toward computing that students from economically marginalized backgrounds bring into computing classrooms, critical computing instruction may better engage with their social, political, and cultural realities, similar to the shift toward more critical digital citizenship education [18], as well as reflect, promote, and center their values, akin to culturally responsive and sustaining computing [41, 61].

6.2 RQ2: What Tensions Might Emerge When Integrating Critical Pedagogy into a Secondary Computing Classroom Context?

Our findings revealed tensions between our participants' computing attitudes, knowledge, self-efficacy, and social consciousness.

We observed a tension between our participants' social consciousness and their computing knowledge in the PLs unit. With their existing social consciousness, our participants identified the assumptions made by designers of technology about user populations, such as assumptions around sight ability and English fluency. However, with limited computing knowledge, they did not critique those assumptions or values beyond a surface-level identification. This observation suggests a need for scaffolding in examining the implicit values and assumptions embedded in technology design, such as the CIDER assumption elicitation technique [50]. This technique has been shown to be successful with post-secondary students but has not been studied with students of younger ages, a potentially fruitful avenue for future exploration.

In addition to a tension between our participants' social consciousness and their computing knowledge, we observed tensions between our participants' social consciousness, and their self-efficacy and attitudes toward technology. Our participants exhibited a rich social consciousness and were aware of the impacts that technology can have on individuals and society more broadly. They identified how the example code to determine financial aid for post-secondary education would exclude many students in need, such as students with larger families or students with disabilities. However, we observed that their socially conscious values did not always translate in their programming design and implementation, whether that be because of a lack of computing knowledge or self-efficacy, or their dispositions toward technology. While our participants identified potential harmful or exclusionary values in the example code, they rarely identified the harmful or exclusionary values embedded in their own solutions. This may be reflective of their self-efficacy, attitudes toward technology, or both. Our participants may have felt confident that they could develop solutions that did not do any harm, reflecting their self-efficacy. Alternatively, this may be reflective of positive attitudes and a resultant trust toward technology. They may have felt that after they addressed possible harmful values reflected in the code, the code could no longer do harm. This finding highlights a need within critical computing instruction to support participants in recognizing and reflecting on their own assumptions, values, and relationships with technology, and consequently, deciding what values and relationships with technology computing should actively promote, as proposed in counter-hegemonic computing [10].

Throughout the course, our participants began to show signs of moving beyond simply knowing of social wrongs to knowing how to address social wrongs [51]. Tensions between our participants' social consciousness, and their computing knowledge and self-efficacy emerged when they attempted to address these wrongs. toward the end of the course, our participants exhibited an awareness of the human values surrounding the creation of technology, such as Bob characterizing the smartness of a computer as inherently dependent on the smartness of the programmer in her end of course reflection. Some participants also expressed ideas on how to address social wrongs perpetuated by technology, such as A & T's final story depicting a re-design of a fictional technology originally designed as inaccessible for common people. This may be evidence for an emergent critical computing consciousness, that some participants may have progressed to the pre-critical or critical stages of sociopolitical development. In contrast, some participants displayed an awareness of possible harmful values embedded in the example code in the *Programming* unit, but expressed practical (i.e., lack of computing knowledge) or perceived (i.e., self-efficacy) difficulties in addressing all the harms. Thus, while our participants displayed signs of computing self-efficacy at the end of the course, it is unclear to what extent they developed a *critical* computing self-efficacy. This result offers some impetus for conceptualizing critical computing self-efficacy; its definition and evaluation would be insightful directions for future research.

Our findings built upon the burgeoning research investigating the applications of critical computing [15, 32, 36, 45, 57] by deepening our understanding of potential challenges that could emerge in practice. These tensions that emerged in our critical computing course highlight the need for scaffolding students in negotiating the complex impacts of technology on their lives, communities, and societies. Vakil and McKinney de Royston's framework of *Youth as Philosophers of Technology* [66] provides some insight. Drawing from pragmatic philosophical inquiry, their framework prioritizes youth learning to engage with the multiple truths and contested meanings of the ethical implications of technology. In this vein, exploring how discoveries from other **Discipline-Based Education Research (DBER)** fields with longer histories of emphasizing critical discourse, such as history, philosophy, and social studies, could apply in critical computing instruction could be a worthwhile direction for future work or iterations of this course. In future iterations of similar courses, we would recommend looking beyond the emerging literature on critical pedagogy in computing, and draw from insights and techniques from other DBER fields, such as Investigative History Teaching [67] or Philosophy for Children [42, 64], to make such interrogation of hegemonic values in computing more accessible.

Beyond these tensions, it is crucial to highlight that despite the instructors' critical approach to teaching computing content and assignments that prompted them to critique technology, most of our participants were still engaged in and were not dissuaded from computing, as evinced in the end of course reflections. In fact, for some of them, the critical framing may have been a vehicle for engagement. For instance, A & T's subversive interpretation of the programming assignment to decide financial aid for post-secondary education tuition offered ideas for a more expansive view of critical computing instruction, one that includes interrogation of both hegemonic and counter-hegemonic values in computing. By interrogating opposing sets of values, critical computing instruction could further reify the idea that technology is inherently value-laden [75], a potential fundamental literacy in critical computing.

Our participants' steady engagement and enthusiasm may be an indication of the resilience of our participants' positive narratives around computing. It may also be due to our practice of teaching that viewed every student as capable of learning computing, regardless of prior experience. We also actively worked to deconstruct our positions of power as instructors, offering vulnerability about our experiences in computing and the role computing has played in our lives and communities. It is also important to note that such engagement with critical content may also be a function of our participants' positionality as people who come from low-income families, would be first-generation university students, are immigrants, or are people of color. A prior implementation of critical computing with students of primarily dominant identities in computing (e.g., White, East/South Asian, men, wealthy) found that while some engaged, others ignored, rejected, or resisted critical content [36].

6.3 Limitations

Our study context and methodological decisions resulted in some threats to both the internal and external validity of our findings. With respect to internal validity, although we took measures to mitigate them, power dynamics between instructors and students are inherent in a formal teaching environment with norms of grades and other forms of evaluation. The positionalities of the instructors may also have impacted what participants were willing to express in their classwork. Although the primary instructor (first author) shared many identities with the participants, she was more than a decade older than the participants, potentially leading them to have viewed her as an authority figure. In contrast, the co-instructor (second author) did not share as many identities, but as an undergraduate researcher, was closer in age and participants may have viewed her as a near-peer. Also, while we analyzed classwork for this study, classwork was only one facet of the learning experience. Class discussions were an important part of the learning experience, where participants were arguably more open in expressing ideas relevant to the study. However, we chose not to capture those discussions because only 7 of the 20 students consented. This limitation reveals a key point of friction in conducting research in critical computing education, wherein the vulnerability that is sometimes required for a critical examination of technology conflicts with research data collection practices. Lastly, the structure of the summer academic program imposed constraints on data collection, especially on the last day of instruction. Our participants were understandably tired of being in an academic program and looking forward to the rest of summer break, impacting engagement in the final reflection at the end of the class. Further, following the last day of instruction was a program-wide showcase of student work in elective courses, which some participants wanted extra time to prepare for.

As for external validity, our participants only represent their own thoughts and values—as shaped by their identities, backgrounds, and experiences—within this specific educational context. Our participants were also perceived to be the most "meritorious" in their schools by their teachers, opted into this elective computing class, and lived in a city with a strong tech industry presence—all of which may have shaped their dispositions toward the course and summer program as a whole. Nonetheless, the socioeconomic backgrounds of the participants, as well as the ethnic and linguistic diversity within the participant group, sheds light on how students of similarly marginalized backgrounds *might* engage in critical computing instruction. It would also be similarly crucial to study how a student population of backgrounds well-represented in computing might also engage in such instruction, because they will likely still form a decent proportion of computing students, as unfortunate as that may be.

Lastly, this study and the textbook upon which it is based [38] were grounded in critical pedagogy, which is not without its limitations. Critical pedagogy was based on Western (e.g., Christian and Marxist) assumptions that sideline indigenous knowledge systems [6]. It would be pertinent to further explore pedagogical approaches in computing that advance non-Western and indigenous *epistemologies*, not just *practices*, such as the works of Archibald et al. [3] and Webster et al. [74].

6.4 Implications and Future Work

In light of our observations, and acknowledging their limitations, our contributions have several implications for both research and teaching. Perhaps most importantly, our findings reveal a need for further study into both competencies and self-efficacy for critical computing. Just as countless efforts in computing education research defined and evaluated the technical skills and self-efficacy of computing, we argue that the ability to critically examine technology's role in society *should* be considered a computing skill. Therefore, it warrants the same, if not more, effort toward researching new pedagogies for developing critical computing literacies, new competencies that not only construct pathways to educational opportunity but are also grounded in the practices of diverse communities and cultures [34], and new ways of assessing its competencies in ways that foster student self-efficacy, but also humility.

Because it was evident that participants brought some awareness of negative impacts, there is also a need for research that examines how to build upon students' existing narratives. Critical framings of computing instruction could start by encouraging students to reflect on both the benefits and perils of technology in their lives and society, using critiques of their own relationships with technology as a catalyst for developing critical computing consciousness. There are numerous open questions for how to do this. Simply communicating alternative narratives about computing, for example, akin to Freire's banking model [17], is unlikely to hold promise [63, 70]. But making space for negative, skeptical, or more humble viewpoints on computing, in light of even marginalized adolescents' fascination with it as a source of empowerment and expression, promises to be challenging. It raises questions about what capacity adolescents have to hold space for conversations about computing and equity, and what rights they have to refuse those conversations [8].

This leaves many open questions for educators, especially in times of increasing legal resistance to even acknowledging narratives that counter notions of color blindness, meritocracy, or the unquestionable good of markets [68]. If the law is suppressing critical consciousness of computing, industry has no interest in funding it, and youth are prone to overlooking critical perspectives when they take on the role of creators, what is the role of educators in resisting these forces? Research will only be part of answering this question; it will likely be educators first, balancing their own visions of computing literacy with those desired by youth. Supporting youth in developing critical consciousness of computing, therefore, is likely to be not only an exercise in responsiveness, but also negotiation.

Acknowledgment

We would like to thank Audrey Le Meur for co-instructing this summer course, and Roseann London and Tony Au for organizing and running the Upward Bound 2022 Summer Academy.

References

[1] StoryCenter. 2024. Retrieved from https://www.storycenter.org/
[2] Adel M. Aladwani and Mohammad Almarzouq. 2016. Understanding compulsive social media use: The premise of complementing self-conceptions mismatch with technology. *Computers in Human Behavior* 60 (2016), 575–581.
[3] Jo-Ann Archibald. 2008. *Indigenous Storywork: Educating the Heart, Mind, Body, and Spirit*. UBC Press.
[4] Ruha Benjamin. 2020. Race after technology: Abolitionist tools for the new jim code.
[5] Cynthia L. Bennett, Burren Peil, and Daniela K. Rosner. 2019. Biographical prototypes: Reimagining recognition and disability in design. In *Proceedings of the 2019 on Designing Interactive Systems Conference*, 35–47.
[6] Chet A. Bowers. 2004. *Re-Thinking Freire: Globalization and the Environmental Crisis*. Routledge.
[7] Derya Ozcelik Buskermolen and Jacques Terken. 2012. Co-constructing stories: A participatory design technique to elicit in-depth user feedback and suggestions about design concepts. In *Proceedings of the 12th Participatory Design Conference: Exploratory Papers, Workshop Descriptions, Industry Cases,* Vol. 2, 33–36.
[8] Angela Calabrese Barton and Edna Tan. 2020. Beyond equity as inclusion: A framework of "rightful presence" for guiding justice-oriented studies in teaching and learning. *Educational Researcher* 49, 6 (2020), 433–440.
[9] Alison Clear, Allen Parrish, Ming Zhang, and Gerritt C. van der Veer. 2017. CC2020: A vision on computing curricula. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, 647–648.
[10] Ron Eglash, Audrey Bennett, Laquana Cooke, William Babbitt, and Michael Lachney. 2021. Counter-hegemonic computing: Toward computer science education for value generation and emancipation. *ACM Transactions on Computing Education* 21, 4 (2021), 1–30.
[11] Ron Eglash, Audrey Bennett, Casey O'donnell, Sybillyn Jennings, and Margaret Cintorino. 2006. Culturally situated design tools: Ethnocomputing from field site to classroom. *American Anthropologist* 108, 2 (2006), 347–362.
[12] Ron Eglash, Juan E. Gilbert, and Ellen Foster. 2013. Toward culturally responsive computing education. *Communications of the ACM* 56, 7 (2013), 33–36.
[13] Virginia Eubanks. 2012. *Digital Dead End: Fighting for Social Justice in the Information Age*. MIt Press.
[14] Virginia Eubanks. 2018. *Automating Inequality: How High-Tech Tools Profile, Police, and Punish the Poor*. St. Martin's Press.
[15] Jayne Everson, F. Megumi Kivuva, and Amy J. Ko. 2022. "A key to reducing inequities in like, AI, is by reducing inequities everywhere first" emerging critical consciousness in a co-constructed secondary CS classroom. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education,* Vol. 1, 209–215.
[16] David Finkelhor, Kerryann Walsh, Lisa Jones, Kimberly Mitchell, and Anne Collier. 2021. Youth internet safety education: Aligning programs with the evidence base. *Trauma, Violence, & Abuse* 22, 5 (2021), 1233–1247.
[17] Paulo Freire. 2018. *Pedagogy of the Oppressed*. Bloomsbury Publishing USA.
[18] Antero Godina Garcia, Sarah McGrew, Nicole Mirra, Brendesha Tynes, and Joseph Kahne. 2021. Rethinking digital citizenship: Learning about media, literacy, and race in turbulent times. *Educating for Civic Reasoning and Discourse* (2021), 319–352.
[19] Patricia Garcia, Germán A. Cadenas, and Kimberly A. Scott. 2023. Expanding theories of sociopolitical development: Centering the intersectional experiences of girls of color in an informal STEM program. *TechTrends* 67, 3 (2023), 407–416.
[20] Nettrice Gaskins. 2023. Interrogating algorithmic Bias: From speculative fiction to Liberatory design. *TechTrends* 67, 3 (2023), 417–425.
[21] Sonia Denise Gatchair. 2007. *Representation and Reward in High Technology Industries and Occupations: The influence of Race and Ethnicity*. Georgia Institute of Technology.
[22] Norma González, Luis C. Moll, and Cathy Amanti. 2006. *Funds of knowledge: Theorizing Practices in Households, Communities, and Classrooms*. Routledge.
[23] Joanna Goode, Julie Flapan, and Jane Margolis. 2018. Computer science for all. In *Diversifying Digital Learning: Online Literacy and Educational Opportunity*, *JHU Press, Maryland*, 45–65.
[24] David Hammer and Leema K. Berland. 2014. Confusing claims for data: A critique of common practices for presenting qualitative research on learning. *Journal of the Learning Sciences* 23, 1 (2014), 37–46.
[25] Ria Hanewald. 2008. Confronting the pedagogical challenge of cyber safety. *Australian Journal of Teacher Education (Online)* 33, 3 (2008), 1–16.
[26] Nathan Holbert, Michael Dando, and Isabel Correa. 2020. Afrofuturism as critical constructionist design: Building futures from the past and present. *Learning, Media and Technology* 45, 4 (2020), 328–344.

[27] Andrew Gary Darwin Holmes. 2020. Researcher positionality–A consideration of its influence and place in qualitative research–A new researcher guide. *Shanlax International Journal of Education* 8, 4 (2020), 1–10.
[28] bell Hooks. 2014. *Teaching to Transgress*. Routledge.
[29] Anne Drew Hu and Aman Yadav. 2023. How K-12 CS teachers conceptualize CS ethics: Future opportunities and barriers to ethics integration in K-12 CS. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education*, Vol. 1, 910–916.
[30] Rahel Jaeggi. 2017. Crisis, contradiction, and the task of a critical theory. In *Feminism, Capitalism, and Critique: Essays in Honor of Nancy Fraser.* B. Bargu and C. Bottici (Eds.), Palgrave Macmillan, 209–224.
[31] Haiyan Jia, Pamela J. Wisniewski, Heng Xu, Mary Beth Rosson, and John M. Carroll. 2015. Risk-taking as a learning process for shaping teen's online information privacy behaviors. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, 583–599.
[32] Britney Johnson, Ben Rydal Shapiro, Betsy DiSalvo, Annabel Rothschild, and Carl DiSalvo. 2021. Exploring approaches to data literacy through a critical race theory perspective. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–15.
[33] Yasmin Kafai, Chris Proctor, and Debora Lui. 2020. From theory bias to theory dialogue: Embracing cognitive, situated, and critical framings of computational thinking in K-12 CS education. *ACM Inroads* 11, 1 (2020), 44–53.
[34] Yasmin B. Kafai and Chris Proctor. 2022. A revaluation of computational thinking in K–12 education: Moving toward computational literacies. *Educational Researcher* 51, 2 (2022), 146–151.
[35] Joe L. Kincheloe and Peter McLaren. 2011. Rethinking critical theory and qualitative research. In *Key Works in Critical Pedagogy.* Kecia Hayes, Shirley R. Steinberg, and Kenneth Tobin. Brill (Eds.), Vol. 32, SensePublishers, 285–326. DOI: https://doi.org/10.1007/978-94-6091-397-6_23
[36] Mara Kirdani-Ryan and Amy J. Ko. 2022. The house of computing: Integrating counternarratives into computer systems education. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education,* Vol. 1, 279–285.
[37] Amy J. Ko, Anne Beitlers, Jayne Everson, Brett Wortzman, and Dan Gallagher. 2023. Proposing, planning, and teaching an equity-and justice-centered secondary pre-service CS teacher education program. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education,* Vol. 1, 583–589.
[38] Amy J. Ko, Anne Beitlers, Brett Wortzman, Matt Davidson, Alannah Oleson, Mara Kirdani-Ryan, and Stefania Druga. 2021. *Critically Conscious Computing: Methods for Secondary Education.*
[39] Amy J. Ko, Alannah Oleson, Mara Kirdani-Ryan, Yim Register, Benjamin Xie, Mina Tari, Matthew Davidson, Stefania Druga, and Dastyni Loksa. 2020. It is time for more critical CS education. *Communications of the ACM* 63, 11 (2020), 31–33.
[40] Gloria Ladson-Billings. 1995. Toward a theory of culturally relevant pedagogy. *American Educational Research Journal* 32, 3 (1995), 465–491.
[41] Gloria Ladson-Billings. 2014. Culturally relevant pedagogy 2.0: Aka the remix. *Harvard Educational Review* 84, 1 (2014), 74–84.
[42] Rotem Landesman, Jean Salac, Jared Ordoña Lim, and Amy J. Ko. 2024. Integrating philosophy teaching perspectives to foster adolescents' ethical sensemaking of computing technologies. In *Proceedings of the 2024 ACM Conference on International Computing Education Research*, Vol. 1, 502–516.
[43] Calvin A. Liang, Sean A. Munson, and Julie A. Kientz. 2021. Embracing four tensions in human-computer interaction research with marginalized people. *ACM Transactions on Computer-Human Interaction* 28, 2 (2021), 1–47.
[44] Jane Margolis, Allan Fisher, and Faye Miller. 2000. The anatomy of interest: Women in undergraduate computer science. *Women's Studies Quarterly* 28, 1/2 (2000), 104–127.
[45] Eric J. Mayhew and Elizabeth Patitsas. 2023. Critical Pedagogy in Practice in the Computing Classroom. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education*, Vol. 1, 1076–1082.
[46] Dan P. McAdams. 2011. Narrative identity. In *Handbook of Identity Theory and Research.* S. J. Schwartz, K. Luyckx, and V. L. Vignoles (Eds.), Springer Science + Business Media, 99–115.
[47] Nora McDonald, Sarita Schoenebeck, and Andrea Forte. 2019. Reliability and inter-rater reliability in qualitative research: Norms and guidelines for CSCW and HCI practice. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–23.
[48] Megan A. Moreno, Katie G. Egan, Kaitlyn Bare, Henry N. Young, and Elizabeth D. Cox. 2013. Internet safety education for youth: stakeholder perspectives. *BMC Public Health* 13 (2013), 1–6.
[49] Karen Mossberger, Caroline J. Tolbert, and Ramona S. McNeal. 2007. *Digital Citizenship: The Internet, Society, and Participation*. MIT Press.
[50] Alannah Oleson, Meron Solomon, Christopher Perdriau, and Amy J. Ko. 2022. Teaching inclusive design skills with the CIDER assumption elicitation technique. *ACM Transactions on Computer-Human Interaction*, 30, 1 (2022), 1–49.
[51] William Paris. 2022. Crisis consciousness, utopian consciousness, and the struggle for racial justice. *Puncta* 5, 4 (2022), 144–166.

[52] Thomas M. Philip, Maria C. Olivares-Pasillas, and Janet Rocha. 2016. Becoming racially literate about data and data-literate about race: Data visualizations in the classroom as a site of racial-ideological micro-contestations. *Cognition and Instruction* 34, 4 (2016), 361–388.
[53] Nichole Pinkard. 2001. Rappin' Reader and Say Say Oh Playmate: Using children's childhood songs as literacy scaffolds in computer-based learning environments. *Journal of Educational Computing Research* 25, 1 (2001), 17–34.
[54] Justin Reich. 2020. *Failure to Disrupt: Why Technology Alone Can't Transform Education*. Harvard University Press.
[55] Steven O. Roberts, Carmelle Bareket-Shavit, Forrest A. Dollins, Peter D. Goldie, and Elizabeth Mortenson. 2020. Racial inequality in psychological research: Trends of the past and recommendations for the future. *Perspectives on Psychological Science* 15, 6 (2020), 1295–1309.
[56] Larry D. Rosen, Alex F. Lim, Julie Felt, L. Mark Carrier, Nancy A. Cheever, Jose M. Lara-Ruiz, Jessica S. Mendoza, and Jeffrey Rokkum. 2014. Media and technology use predicts ill-being among children, preteens and teenagers independent of the negative health impacts of exercise and eating habits. *Computers in Human Behavior* 35 (2014), 364–375.
[57] Jean J. Ryoo, Alicia Morris, and Jane Margolis. 2021. "What happens to the raspado man in a cash-free society?": Teaching and learning socially responsible computing. *ACM Transactions on Computing Education* 21, 4 (2021), 1–28.
[58] Henrik Skaug Saetra. 2023. *Technology and Sustainable Development: The Promise and Pitfalls of Techno-Solutionism*. Taylor & Francis.
[59] Jean Salac, Alannah Oleson, Lena Armstrong, Audrey Le Meur, and Amy J. Ko. 2023. Funds of knowledge used by adolescents of color in scaffolded sensemaking around algorithmic fairness. In *Proceedings of the 2023 ACM Conference on International Computing Education Research*, Vol. 1, 191–205.
[60] Linda J. Sax, Jennifer M. Blaney, Kathleen J. Lehman, Sarah L. Rodriguez, Kari L. George, and Christina Zavala. 2018. Sense of belonging in computing: The role of introductory courses for women and underrepresented minority students. *Social Sciences* 7, 8 (2018), 122.
[61] Kimberly A. Scott, Kimberly M. Sheridan, and Kevin Clark. 2015. Culturally responsive computing: A theory revisited. *Learning, Media and Technology* 40, 4 (2015), 412–436.
[62] Arlene Skolnick. 1975. The limits of childhood: Conceptions of child development and social context. *Law & Contemporary Problems* 39 (1975), 38.
[63] Laura A. Taylor and Michiko Hikida. 2020. Unpacking everyday critical pedagogy: Languaging critique and dialogue. *Literacy Research: Theory, Method, and Practice* 69, 1 (2020), 266–284.
[64] Steve Trickey and Keith J. Topping. 2004. 'Philosophy for children': A systematic review. *Research Papers in Education* 19, 3 (2004), 365–380.
[65] Sepehr Vakil. 2018. Ethics, identity, and political vision: Toward a justice-centered approach to equity in computer science education. *Harvard Educational Review* 88, 1 (2018), 26–52.
[66] Sepehr Vakil and Maxine McKinney de Royston. 2022. Youth as philosophers of technology. *Mind, Culture, and Activity* 29, 4 (2022), 336–355.
[67] Bruce A. VanSledright. 2010. *The Challenge of Rethinking History Education: On Practices, Theories, and Policy*. Routledge.
[68] Adriana Villavicencio, Dana Conlin, and Olga Pagan. 2023. Practice partnerships in pursuit of racial justice in schools: Navigating a hostile sociopolitical climate. *Educational Policy* 37, 1 (2023), 250–275.
[69] Emily A. Vogels and Risa Gelles-Watnick. 2023. Teens and Social Media: Key Findings From Pew Research Center Surveys. Retrieved from https://www.pewresearch.org/short-reads/2023/04/24/teens-and-social-media-key-findings-from-pew-research-center-surveys/
[70] Shirin Vossoughi and Kris D. Gutiérrez. 2016. Critical pedagogy and sociocultural theory. In *Power and Privilege in the Learning Sciences*. Indigo Esmonde and Angela N. Booker (Eds.), Routledge, 157–179.
[71] Roderick J. Watts and Constance Flanagan. 2007. Pushing the envelope on youth civic engagement: A developmental and liberation psychology perspective. *Journal of Community Psychology* 35, 6 (2007), 779–792.
[72] Roderick J. Watts, Derek M. Griffith, and Jaleel Abdul-Adil. 1999. Sociopolitical development as an antidote for oppression—theory and action. *American Journal of Community Psychology* 27, 2 (1999), 255–271.
[73] Roderick J. Watts, Nat Chioke Williams, and Robert J. Jagers. 2003. Sociopolitical development. *American Journal of Community Psychology* 31, 1–2 (2003), 185–194.
[74] Emma Webster, Allan Hall, Yvonne Hill, Cecil See, Eunice Simons, Rachael Havrlant, and Regina Osten. 2022. Building cultural responsiveness in a mainstream health organisation with '8 Aboriginal Ways of Learning': A participatory action research study. *Australian and New Zealand Journal of Public Health* 46, 4 (2022), 517–523.
[75] Langdon Winner. 2017. Do artifacts have politics? In *Computer Ethics*. John Weckert (Eds.), Routledge, 177–192.
[76] Aman Yadav, Marie Heath, and Anne Drew Hu. 2022. Toward justice in computer science through community, criticality, and citizenship. *Communications of the ACM* 65, 5 (2022), 42–44.

Received 7 July 2023; revised 4 September 2024; accepted 17 October 2024