Designing Desired Support for Learning Programming with Minoritized Women Students in Computing

Xinying Hou
University of Michigan
Ann Arbor, MI, USA
xyhou@umich.edu

Evie Katmanivong
University of Michigan
Ann Arbor, MI, USA
ekatmani@umich.edu

Xu Wang
University of Michigan
Ann Arbor, MI, USA
xwanghci@umich.edu

Barbara J. Ericson
University of Michigan
Ann Arbor, MI, USA
barbarer@umich.edu

Abstract

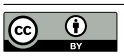
Background and Context. There are growing efforts to increase the percentage of secondary students who take computing courses. However, U.S. women identifying as Black/African American, Hispanic/Latina, and/or Native American remain underrepresented in computing and face persistent challenges in learning to program. Moreover, it remains underexplored what types of programming learning support minoritized high school women desire.

Objectives. To address this gap, this study investigates the challenges minoritized high school women encounter in current help-seeking practices and explores the forms of programming learning support they desire.

Method. We conducted design sessions with 12 high school women who are minoritized in computing and enrolled in Advanced Placement Computer Science courses. Using thematic analysis, we examined the challenges they encountered with existing help-seeking resources, as well as their preferred format and support features.

Findings. Students faced several challenges in their current help-seeking practices, including discomfort seeking help from others, unclear problem instructions, and insufficient resources, such as support resources that were timely but inaccurate, or valid but impractical. From students' designs and experiences, we identified three key types of desired support content (explanatory information, code examples, diagnostic information), each with specific characteristics. We also highlighted five high-level patterns of desirable resources, including visible contributions to solving issues, thoughtful encouragement, links to trusted helpers, the promotion of self-reflection, and the fostering of strategy.

Implications. Beyond specific resource preferences, our findings also suggest several twofold high-level needs in desired programming learning support. These results may inform multiple stakeholders. For example, instructors can use them to refine instructional materials and iterate communication styles that better support students. Additionally, educational tool researchers and developers can use these insights to design more effective, responsive programming support tools that address nuanced student needs.



ICER 2026 Vol. 1, Uppsala, Sweden

ACM ISBN 979-8-4007-2203-5/26/08
https://doi.org/10.1145/3765964.3811666

CCS Concepts

• **Human-centered computing**; • **Social and professional topics** → **K-12 education**; **Computing education**;

Keywords

Advanced Placement, K-12 Education, Programming Learning, Computing Education, Help-seeking

ACM Reference Format:
Xinying Hou, Evie Katmanivong, Xu Wang, and Barbara J. Ericson. 2026. Designing Desired Support for Learning Programming with Minoritized Women Students in Computing. In *Proceedings of the ACM Conference on International Computing Education Research Vol.1 (ICER 2026 Vol. 1), August 11–14, 2026, Uppsala, Sweden.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3765964.3811666

1 Introduction

Improvements in access to foundational computer science (CS) courses are emerging at the K–12 level in the United States [21]. Based on 2025 data, 60% of U.S. high schools offer at least one foundational CS course, and 12 states have passed legislation requiring every student to complete at least one CS course before graduating from high school [21]. Another key benchmark for evaluating equity in U.S. K–12 computer science education is the representation in Advanced Placement Computer Science (AP CS) exams. AP CS courses are college-level computer science courses offered in high schools. After completing these courses, students may take the AP CS exams, which are standardized assessments administrated annually in May [1]. Student engagement in AP CS exams can mark an important milestone in their future computing pathways, including sustained interest and success in later computing courses [9, 125]. According to data from the year-over-year AP CS exam published by Code.org [22], the overall exam participation has increased steadily since 2010, and the gender gap has been narrowing over time [22]. However, in the most recent exam data (2024), participation rates remain uneven across race/ethnicity and gender. Black female students account for 3% of exam takers, compared to 4% for Black male students. Hispanic/Latina females represent 6%, while Hispanic/Latino males account for 12%. In contrast, White and Asian male students together account for 44% of AP CS exam takers. This participation gap persists throughout high school and

[1] https://apstudents.collegeboard.org/courses/ap-computer-science-program

extends into postsecondary education and related career opportunities [89, 103, 124]. Since students' early computing experiences can serve as gateways to future computing trajectories [6, 49, 104], understanding *the programming learning support desired by minoritized high school women* is critical to supporting their continued participation in computing.

Therefore, this work aims to design *with* minoritized high school women in computing to better understand their needs and explore this design space [88]. Through this approach, we aimed to derive design implications grounded in their experiences and centered on the support they value. We conducted design sessions with 12 high school women who enrolled in AP CS courses during the 2024-2025 academic year and were racial/ethnic minorities in computing (Black/African American, Hispanic/Latina, and/or Native American) [26, 106]. These design sessions aimed to prioritize the needs and perspectives of these minoritized women students, recognize their existing assets, and ensure their involvement in shaping desired learning support solutions [14, 20, 28]. We address the following research questions, focusing on the challenges, specific design dimensions, and the high-level criteria applied to determine the desired support:

- RQ1: What challenges do minoritized women students face when seeking support while learning to program?
- RQ2: What are minoritized women students' desired supports for learning to program?
- RQ3: What matters to minoritized women students when determining desired support for learning to program?

Our findings show that minoritized high school women faced several barriers to accessing high-quality programming learning support, including discomfort seeking human help, unclear task instructions, the need to apply specific strategies to get help, and poor-quality resources (RQ1). Drawing on their self-designed supports, we identified three types of desired support content types, each with distinct characteristics (RQ2), and we summarized students' high-level values and concerns regarding specific support features (RQ3). As such, this work made three contributions to future computing education research:

- (1) We extend prior help-seeking research in computing education by providing empirical findings about minoritized high school women's challenges when seeking help in introductory programming learning.
- (2) We highlight design dimensions for providing desirable programming learning support to minoritized high school women in computing.
- (3) We uncover the high-level considerations that minoritized high school women value in support resources while learning programming.

2 Related Work

2.1 The intersectionality of minoritized women in computing education

Concerns about participation and representation remain ongoing in computing education [84]. Intersectionality provides a framework for understanding the compounded barriers minoritized women face in computing, as they experience double disadvantages at the intersection of race/ethnicity and gender identities [80, 92, 96]. The intersectionality framework reveals that personal identities do not occur in separation, but intersect and co-construct one another [7, 32].

Prior research used this intersectional lens to examine the experiences of minoritized women in computing, and showed that their pathways are shaped by the combined influences of race and gender [36, 97, 132]. For example, Black women often face distinct challenges and adopt unique strategies in computing, which sets their experiences apart from those of Black men and non-Black women [97, 132]. Specifically, Thomas et al. (2018) interviewed 11 Black women in computing and found that they faced isolation, self-doubt, resistance, stereotype-based expectations, and had to employ specific strategies to sustain their presence in the field. Similarly, Rankin et al. (2021) interviewed 24 Black women and argued that computing education is a matrix of intersecting oppressions for Black women, from cold temperatures in their first K-12 programming class to stereotypes in computing company cultures. Later research extended this line of work into secondary education. They identified factors that contribute to feelings of marginalization as well as factors that support persistence [36].

In the context of equity-focused interventions, Fisk et al. (2024) quantitatively examined the effectiveness of different interventions in helping Black women to persist in computing. They found that career awareness and faculty collaboration supported Black women's persistence intentions in computing, and noted that many efforts to broaden computing participation failed to focus on racially minoritized women [33]. Intersectionality is also gaining increasing attention when designing technologies [30, 106, 131], generating a rethinking of how technologies are created for underserved populations. For instance, Solyst et al. (2023) applied the framework of culturally responsive computing and explored how Black girls envisioned fair AI to facilitate learner-centered AI literacy programs for them, an underrepresented group in AI development. Recently, Li et al. (2025) also investigated how Black girls participated in a culturally responsive computing summer camp and found that the girls began to see themselves as technology creators as the camp moved forward. In addition to these interventions and explorations, more research is needed to explore the support that minoritized high school women desire in formal programming classes.

2.2 Students' help-seeking in the context of programming education

Help-seeking has been a frequently studied topic in programming education [53, 54] and is recognized as a key self-regulated learning strategy [45, 46]. There are multiple academic help-seeking frameworks available, including the five stages of problem-solving help-seeking in classroom settings [76], help-seeking in interactive learning environments [3], and the eight-stage model for broader academic help-seeking [46].

In the context of programming education, there are multiple potential help sources available, including formal sources (textbooks [10], teaching assistants [95], and official manuals [79]) and informal sources (peers [41], online-searching [40] and specialized Q&A platforms [69]). For example, Doebling and Kazerouni (2021)

investigated undergraduate computing students' help-seeking behaviors and revealed a common progression where students began with easily accessible but low-quality sources (such as peers and online) before turning to less accessible but higher-quality sources (such as teachers) [29]. Studies also found that women used formal sources more [29], chose to post anonymously in forums [42, 61], and waited more patiently for office hours [35] than male students. Lately, Ko et al. (2025) revealed that gender-minority and less confident students sought help from internal sources, while gender-majority students favored external sources. Also, Latinx/e/a/o students relied on people outside of the course more [54].

With the rise of emerging AI technologies, scholars have increasingly turned their attention to AI-mediated help-seeking. For instance, Hou et al. (2024) found that while these AI models are rapidly adopted, they have not yet replaced traditional help resources. Some key motivations to seek help from generative AI tools include rapid iteration and avoiding social pressures. Similarly, another study examined the factors motivating students to use web resources and GenAI in the classroom. Results showed that students' perceived level of difficulty of using GenAI tools shaped their intention to adopt them [111]. After analyzing student-LLM conversation interactions in an AI course, McNichols et al. (2026) reported that students who turned to LLMs for conceptual understanding and coding support tended to perform better. Regarding desirable characteristics for AI assistants in programming learning, Denny et al. (2024) found that students valued tools with immediate and engaging support, and features that allow them to maintain autonomy over their learning.

Beyond these common help sources, researchers and computing education tool designers have developed various instructional and technology-based supports for different programming learning contexts. For instance, Cunningham et al. (2021) proposed a scaffolding mechanism intended mainly for college-level conversational programmers. Tang et al. (2025) developed a system to generate personalized feedback on in-class coding tasks, though their focus was also at the college level. Efforts to support K–12 programming learning are relatively underexplored and concentrated mainly in block-based environments. For example, Chen et al. (2024) created an AI-augmented system to support autonomous programming learning for middle schoolers, but within the block-based programming context. Although Limke et al. (2025) and Limke et al. (2024) examined design needs for online programming learning systems, they did so from the perspective of K–12 teachers.

The needs and desired programming-learning support of minoritized high school women remain largely unaddressed. This work seeks to fill this gap by developing a deeper understanding of these factors.

2.3 Barriers to high-quality help in secondary computing education

Unlike college computing, pre-college computing often lacks sufficient, high-quality help sources. Reasons include low school investment [55, 82, 123], a shortage of qualified teachers [62], and uneven access to computers at home [105]. At the institutional level, computing education is not viewed as a primary priority in many K-12 schools and districts [123], reflected in insufficient investments such as limited hardware and software [55, 82]. Therefore, familiar college-level help sources, such as weekly office hours led by teaching staff [51], can be uncommon in high schools.

While teachers could serve as a primary source of high-quality help, in the U.S., there are not enough qualified computing teachers for computing education in secondary schools [15, 60, 78, 109]. Many K-12 computing teachers may not hold a degree in computer science [55, 62], and they face challenges in supporting students' understanding [107]. Teacher demographics can also contribute to a sense of disconnection for underrepresented students. A nationwide teacher survey showed that most high school CS teachers are white, while teachers of color remain minoritized [55]. Additionally, technology infrastructure, such as computers and internet access, is a prerequisite to accessing online help resources (e.g., AI, social media platforms, and electronic textbooks). A 2023 survey of high school students who registered for a national standardized test found that 87% of students have access to laptop computers at home. However, Black and Hispanic students reported less computer access at home than White and Asian students [105].

In addition, high school students themselves have other non-CS courses and test requirements, limiting the time they can devote to computing learning [15]. Regarding help-seeking patterns among high school computing students, Cheong et al. (2004) revealed that high school African American students tended to seek help more, and female studens were more likely to have more useful and adaptive help-seeking behavior (instrumental help-seeking). However, fewer studies have examined the challenges minoritized high school women face in seeking help while learning to program. Our work aims to address this.

2.4 Design educational solutions with students

Designing educational solutions with stakeholders can leverage their expertise to create solutions that are better aligned with their needs [57, 94] as well as more contextually relevant and practical [28, 102]. Students, with their extensive experience with educational activities, can readily recall and elaborate on their prior problems and needs, propose solutions, and generate diverse ideas [119]. Designing with students can also help students develop ownership and agency to shape their learning. Prior design research has explored designing a range of educational solutions with students [88]. One line of research explores the integration of new technologies in education with students. Within college-level contexts, Prasad et al. (2025) held co-design workshops with undergraduates to explore prototypes of future tools with multimodal large language models to deal with educational problems. Similarly, Zheng et al. (2024) designed with undergraduates to explore the future of project-based learning with AI. In secondary education, Shamamyan et al. (2025) designed with middle school students about trustworthy virtual peer agents for math learning.

Beyond looking into the boundaries of new technology and education, prior researchers have collaborated with students from specific groups to address their existing educational challenges. For instance, Fan et al. (2025) designed with blind and low-vision students to improve their access to statistical instructional practices. In addition, Cho et al. (2019) designed with low-income Latinx families to connect them with affordable out-of-school learning resources. In

computing education, Anthraper et al. (2024) designed with transfer students in computing majors for a peer-mentoring learning analytics system to enhance their academic and social engagement. With a similar overarching goal of addressing the challenges that minoritized women students face in learning programming, this work conducted design sessions *with* minoritized high school women to understand their desired programming learning support, leading to recommendations for future educational design and initiatives. This study centers on the students with firsthand experience in AP CS, as their lived experience within this learning context can provide a grounded basis for informing designs that address their needs [101].

3 Methods

We conducted design sessions with 12 U.S. high school students who had participated in AP CS courses and self-identified as minoritized women in computing. All design sessions were conducted virtually through Zoom with digital whiteboards.

3.1 Participants and Recruitment

Following IRB (Institutional Review Board) approval, students were recruited through two particular routes. This included AP CS teacher groups as well as a free online support program for AP CS students. To be eligible for this study, students needed to self-identify as a woman from a range of gender identity options (e.g., man, nonbinary) and as a member of one or more racial or ethnic groups underrepresented in computing (e.g., Native American / Alaska Native, Hispanic / Latina, and/or Black / African American). Students were also required to be enrolled in an AP CS course, such as AP CS Principles or AP CS A. Design sessions were conducted after obtaining student assent or consent (for those 18 and older), parental or guardian consent (for those under 18), and completing pre-surveys about programming learning experiences. Students were asked to share their screens during the design sessions, which were recorded for analysis. After the design sessions, all participants were referred to by unique de-identified IDs in the analyses and reports (from PID01 to PID12 in this work). Upon completion, each participant received a $25/hour USD gift card. A total of 12 students finished the design session [16]. Basic information for each student can be found in Table 1. All students in our study reported having access to computers outside of class.

3.2 Study Procedure and Activities

3.2.1 Main Procedure. Each design session lasted around 60 minutes and was conducted on a web-based digital whiteboard. The session included three main parts (Figure 1 and Figure 3 show the sample worksheets we used). (1) Reflect and Situate: A comic book–style template was designed for students to fill in with one positive and one negative instance from their prior programming problem-solving and help-seeking experiences (Figure 1-left). (2) Ideate and Design: After situating into two specific scenarios, students were guided to ideate what desired support they want to receive in two scenarios, how it would function, and the expected outcomes with possible adaptations (Figure 1-right). To support students who needed help brainstorming, we later provided five common help providers (e.g., teacher, peer/friend/classmate, computer, AI, and role model) as potential starting points for their design. A student example can be found in Figure 2. (3) Select and Rank: After completing their own designs, we presented students a board that listed help resource candidates, shown conceptually in text form (Figure 3-left). We walked students through each of them and asked for their opinions on whether to keep or remove them, along with their reasoning. We then asked students to select their top three resources from the candidate board and rank them with the two supports they designed, explaining their choices (Figure 3-right). To prevent students from being influenced by the provided resource candidate board, the board was hidden until they had completed the first two parts. During each step, students could move, select, type, or draw on the worksheets. For students who preferred to speak, researchers entered the text based on the students' spoken input, which the students then reviewed and verified. Researchers also conducted follow-up interview questions to explore the students' choices, perceptions, and design considerations. Example interview questions included *"why do you think that could be ideal support compared to the current one?"* and *"why you don't like this feature?"*.

3.2.2 Board for Resource Candidate Selection. The activity in Part 3 was inspired by the UX speed dating approach [24]. By walking through, comparing, and making decisions about diverse conceptual support resource ideas, this activity aimed to gain insight into what matters to minoritized women students when determining desired support for learning programming. Building on existing help-seeking literature in computing education [29, 72, 73, 86, 133], two researchers iteratively brainstormed and designed a set of resource candidates across five main categories (Figure 3): (1) knowledge-centered resources (such as *"explanations on the involved concepts"* and *"link to the related textbook materials"*); (2) metacognition and motivation (such as *"encouraging messages to continue"*, *"suggestions to take a break and then come back"*, and *"information on how many other students are stuck at the same point before"*); (3) AI applications (such as *"an AI chatbot can only communicate about this problem"*); (4) access to specific help providers (such as *"get help from a teacher"* and *"get help from a classmate/friend"*); and (5) community forums (such as *"an anonymous forum where you can share your difficulties with other classmates"*). Some candidates may include placeholders in the description for students to add more specific details if they wish, such as *"a forum to discuss the problem with [...]"*. Through a broad range of potential support resources, this activity aimed to capture students' high-level values and concerns to inform the design of future support.

3.3 Data Analysis

The analysis focused on both the design artifacts and conversations during sessions. We conducted reflexive thematic analysis on the data to answer the research questions. This method was chosen for both its flexibility and depth [11]. Its reflexive approach enabled researchers to actively construct meaning from the data, while acknowledging that their interpretations were influenced by their own experiences [12, 13]. Two researchers collaborated to analyze the data. To address different research questions, they iteratively analyzed data on students' current support practices, their designs,

Table 1: Student Basic Information (In the U.S., high school typically covers grades 9 to 12)

PID	Gender	Race/Ethnicity	Language at Home	Programming experience before this AP CS class
P01	Woman	Black or African American, Asian or Pacific Islander, Native American or Alaska Native	Only English	> 12 months
P02	Woman	Hispanic or Latino (Latina)	Only a language other than English	> 12 months
P03	Woman	Black or African American	Only English	7-9 months
P04	Woman	Black or African American	Only English	9-12 months
P05	Woman	Black or African American	Only English	4-6 months
P06	Woman	Hispanic or Latino (Latina)	Primarily another language, secondary English	> 12 months
P07	Woman	Black or African American, Hispanic or Latino (Latina)	Only English	9-12 months
P08	Woman	Black or African American	Bilingual (English & Another language)	1-3 months
P09	Woman	Black or African American	Primarily English, occasionally another language	> 12 months
P10	Woman	Black or African American	Only English	9-12 months
P11	Woman	Black or African American	Only English	1-3 months
P12	Woman	White, Black or African American, Hispanic or Latino (Latina)	Primarily English, occasionally another language	7-9 months

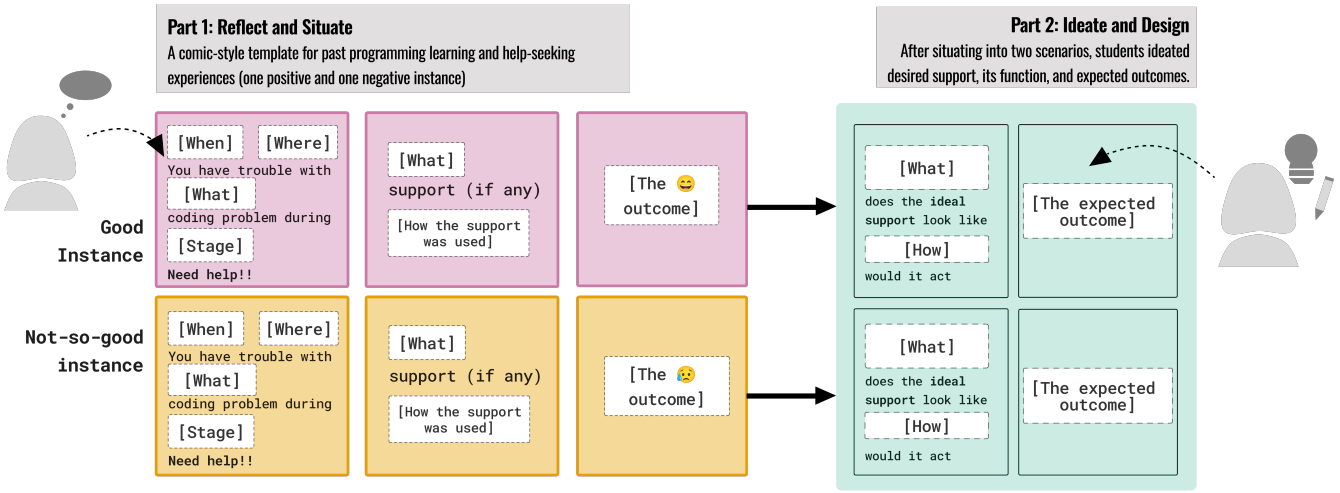


Figure 1: Sample worksheets given to students (Parts 1 and 2). In Part 1, students reflected on past experiences and filled in two comic-style templates (one positive and one negative). It covers when, where, what task, the problem-solving stage that needed support, what support was obtained, how it was used, and the outcome. In Part 2, based on the two scenarios from Part 1, students ideated and designed the desired support, how it would work, and the expected outcome.

and their values regarding potential support resources while learning to program. For each part, they first went through the data independently and applied open coding to identify key codes. They then met to discuss and iteratively developed the final themes. The codes were iteratively refined, integrating inductive codes that were summarized from the data with deductive structures from RQs. During discussions, the two researchers collaborated to build shared understandings into themes. After analyzing each RQ individually, they also explored potential thematic overlaps across them. This ensured that the final themes reflected a comprehensive interpretation of the data. We report our findings in subsection 4.1, subsection 4.2 and subsection 4.3.

4 Findings

4.1 RQ1: Challenges faced by minoritized women students in seeking programming learning support

Students' self-reported comfort with seeking help in their current APCS course and their frequency of using different help sources

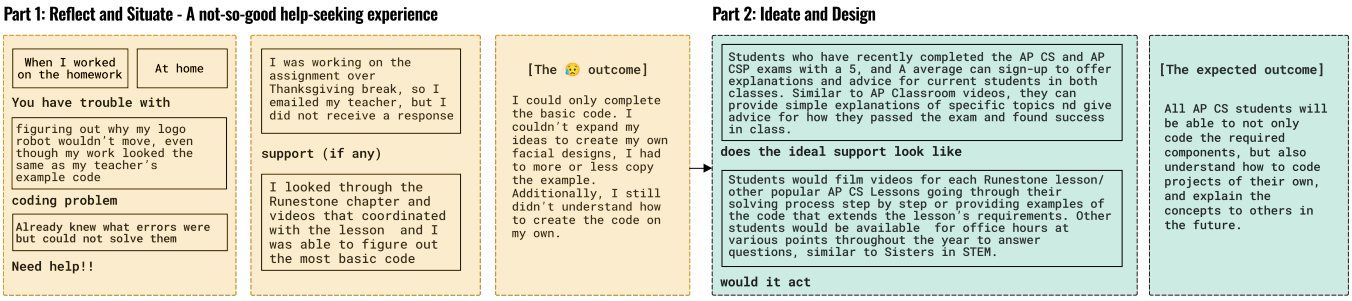


Figure 2: In this example, P12 described a negative help-seeking experience when she had trouble with homework over break. When the teacher did not reply, she was able to figure out some basic code from other resources, but could not complete all of it, and still did not fully understand how to code it independently (Part 1). She proposed a support design centered on students who had passed the APCS exam with high scores, suggesting they record video problem-solving tutorials, hold regular office hours, share personal learning strategies, and cover advanced computing topics. For the expected outcome, she hoped all APCS students would be able to meet requirements, finish programming projects independently, and explain programming concepts to others (Part 2).

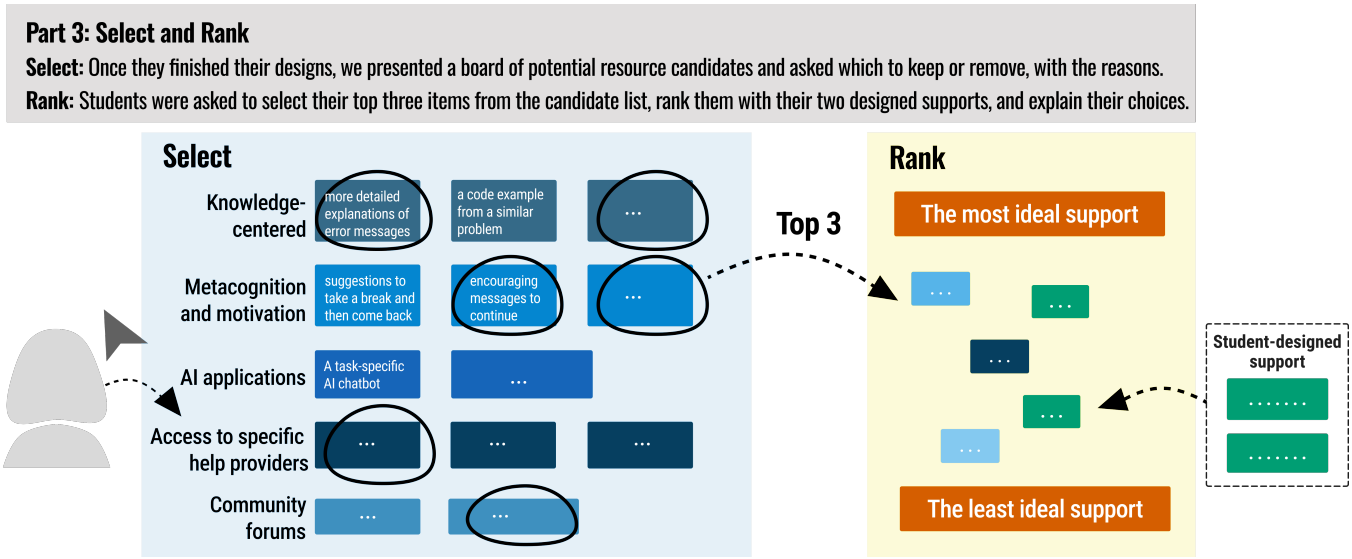


Figure 3: A sample worksheet (Part 3). The board for resource candidate selection was hidden until students completed their designs in Part 2. Afterward, students walked through each support resource candidate, decided which to keep or remove with reasons, then selected their top three and ranked them with the two they designed in Part 2.

(*"How often do you access the following resources for this APCS course?"*) are reported in Table 2. It is worth noting that the availability of the help types listed in Table 2 may have varied across students, though this was not captured in our data. For RQ1, we report findings from students' reflections on current practices, especially their past negative help-seeking experiences. Each theme highlights a challenge that students reported facing when seeking help during programming learning, with student counts showing how widely it was shared. It includes limited support resources, a conflict between speed and reliability, valid but impractical content, strategies for smoother help-seeking, and vague instructions. We ordered the challenges by the number of students who mentioned them.

Not enough support resources available (7 of 12). A major challenge was the lack of sufficient resources, especially when the materials provided by the teachers were unhelpful. For example, when P05 prepared for a quiz at home, she had no textbook on the topic and failed that quiz. P02 also explained her experience as *"but I didn't really get help, other than spending a couple of hours trying to fix it."* Similarly, P07 described a related challenge: *"There weren't enough resources, so I didn't know of what resources I could use...So the only knowledge I have is like the fundamentals."* (P07)

The conflict between speed and reliability (6 of 12). For secondary school students, teachers and peers are their primary everyday resources in academic settings [4, 99]. Students can also make use of online resources and AI tools. In line with previous

Table 2: Students' self-reported general help-seeking comfort and the frequency of accessing common help-seeking sources in the current CS course (– indicates students reported N/A.)

PID	Help Comfort[1]	Instructor (In-class)	Instructor (Online)	Peers (Same-class)	Peers (Other)	Textbook	AI	Online Materials (Course-specific)	Online Materials (Other)	Specific Resources Named[2]
P01	Rarely	Most of the time	Sometimes	Always	Rarely	Sometimes	Most of the time	Most of the time	Rarely	AI
P02	Most of the time	Most of the time	Never	Sometimes	Never	Never	Never	Rarely	Rarely	YouTube, CodeHS, Runestone Academy
P03	Most of the time	Most of the time	Most of the time	Sometimes	Sometimes	Sometimes	Most of the time	Sometimes	Sometimes	Notes and Previous Labs
P04	Rarely	Rarely	Never	Most of the time	Sometimes	Sometimes	Always	Most of the time	Always	TikTok, YouTube, ChatGPT, Khan Academy
P05	Sometimes	Sometimes	Sometimes	Rarely	Never	Never	Most of the time	Most of the time	Sometimes	Youtube videos
P06	Always	Sometimes	Rarely	Most of the time	Rarely	Never	Never	Sometimes	Rarely	-
P07	Always	Always	Rarely	Rarely	Never	Never	Never	Never	Never	-
P08	Sometimes	Always	Rarely	Sometimes	Rarely	Sometimes	Never	Sometimes	Sometimes	Stack Overflow, YouTube, LeetCode, GeeksforGeeks
P09	Sometimes	Sometimes	Never	Sometimes	Never	Rarely	Never	Always	Sometimes	W3Schools, Google Search
P10	Always	Never	Sometimes	Rarely	Never	Most of the time	Rarely	Always	Rarely	YouTube, Eimacs, Reddit
P11	Most of the time	Most of the time	Rarely	Sometimes	Never	Sometimes	Never	Never	Never	-
P12	Sometimes	Always	Rarely	Sometimes	Sometimes	Sometimes	Sometimes	Sometimes	Sometimes	Textbooks

[1] The related survey question: *"When you need help in this APCS course, how comfortable are you asking for help?"*
[2] The related survey question: *"What other resources do you use when you need help in this APCS course?"*

impressions of help sources [29], students often described peer and online support as immediate but potentially inaccurate (P01, P06, P10): *"I also tried to use Google AI, it did not work. So I also use my friends ... He helped kind of, not really. I mostly must have struggled through it ... I think I just submitted it and dealt with the penalty points."* (P10) In contrast, teacher support is often more reliable but harder to access. In-class teacher availability could be limited because of large class sizes (P03): *"Thirty kids in my class, so he (the teacher) had to help other kids. So I didn't want to take away too much time from him."* (P03). Also, it was hard to reach teachers outside of class (P11), and the teacher quality can vary (P05, P10): *"the teacher didn't teach us necessarily how it worked."* (P05).

Valid but impractical content for solving struggles (5 of 12). Even when support was provided, it could be objectively valid, but not truly helpful in addressing students' specific issues. Such valid but confusing content can come from teacher-provided materials (P05) or teachers' clarifications (P03, P04). For instance, while working on the homework in class, P03 already recognized her errors, but did not know how to fix them. The teacher offered some clarifications, but she could not apply them due to a lack of understanding: *"I tried asking my teacher, but I didn't really understand his approach to doing it...He gave some advice, but I did not know how to approach it in my code."* (P03) Additionally, students may receive untargeted resources, which required considerable time to interpret, but ultimately were not helpful: *"The resource and function my teacher gave me didn't provide me with any help for the type of game I was creating. I had to find my own resources and mix many different functions and code together to create my own code."* (P09)

Beyond the content quality itself, another problem is that the support, though accurate, was too disengaging to be useful. P12 utilized Varsity Tutors to help her learn Java, but found herself unable to focus on the content, leading to short memorization rather than real learning: *"As far as Varsity Tutors, all the tutors that I was assigned to were very monochromatic ... the way they spoke was very, almost robotic, very dull. And I couldn't necessarily focus on what they were saying ... It didn't engage my brain to remember the information."* (P12)

Strategies required to make help-seeking smoother (3 of 12). Due to the general teacher-student hierarchy, some students spoke about the pressure they felt when approaching teachers to ask questions (P01, P05). This pressure could be reduced if the student felt positively connected to the teacher or held the confidence to not be impacted by the teacher's impressions. P01 believed that teachers liked her and, without concern for how they saw her, often went directly to ask questions: *"I would like to understand it more, so I'll just go up to them and ask them. And if they get mad at me, oh, so what? Like, oh well, it's just helping me learn better. So sometimes if like a student doesn't want to ask the question The other classmate or me can do it for them too."* (P01) She also shared this **deliberate asker selection** as a common strategy when trying to get teachers' support: *"we kind of know when a teacher favorites somebody a little more than another person, so we'll send them up to ask the question."* (P01)

In contrast, P05 expressed general concerns about asking teachers questions: *"Mostly because I feel more comfortable going to a peer than a teacher."* and acknowledged her general discomfort in asking

for help to humans: *"I think a lot of times when I fail something, it's because I didn't feel like I had this faith to ask for clarification."* (P05)

Ambiguous instructions undermine the usefulness of other support (1 of 12). A prerequisite for other support to be practical is precise and detailed task instructions. Otherwise, it can lead to unsuccessful support experiences, even when other help sources were available. This was even more essential when the instructor had specific expectations. P06 explained how an ambiguous instruction led to issues even though she asked her peers to check and spent a long time working on it, *"If the instructions were clear about the project and how our teacher wanted us to do it, (it) would make it a lot easier to understand the logic of what the project is asking us to do."* (P06)

4.2 RQ2: Design dimensions of desired programming learning support: what, how, and who?

We then explored the desired support these minoritized women students seek during problem-solving. Three specific content types emerged as desired support (**what**). These include *explanatory information* (P01, P02, P03, P05, P06, P11, P12), *code examples* (P02, P04, P05, P07, P08, P11), and *diagnostic information* (P03, P10, P11, P12). However, simply having such content does not solve student struggles; its effectiveness depends on the way the content is formulated (**how**), which is also inherently tied to **who** delivers it. The desired characteristics of effective support information formulation are shared rather than conflicting, and focusing on only one is often not enough.

4.2.1 Explanatory information as support (7 of 12): Expert-level credibility, Peer-level tone, Room for independent thinking. Explanatory information helps address confusion and supports a more in-depth understanding, especially in one-on-one settings, where students can freely ask follow-up questions until they fully understand. In their current practices, students typically obtain explanatory information from teachers (P01, P03, P06, P07, P09), peers (P01, P03, P06), tutors (P08), and online resources (P05, P10). Our findings suggest that effective explanatory information may not originate from a single source, but from the integration of complementary perspectives.

Peer-level tone for easier comprehension (4 of 7). When ideating their support design, four students (P01, P05, P08, P11) emphasized the importance of obtaining explanatory information from peers. Because they shared similar perspectives and experiences and communicate at the same level, it became easier to make sense of the content: *"I think it's just like the way that they (friends) talk, it just goes to our brain a little easier."* (P01). P08 selected peer as the ideal support provider with a similar reason: *"They (peers) would know the topic and they could explain it to me, so that I would be able to comprehend what the question or problem is asking me to do."* (P08)

By contrast, while teachers have greater expertise and can often provide more accurate explanations, sometimes their clarification could be over-complicated. In this case, it could be hard for students to digest: *"Because it's better to get a kid's perspective on something, because you most likely will think more similarly. I feel like teachers sometimes, when you don't understand things, they'll overexplain it, and it'll become even more confusing."* (P01)

Calibrate expert view to land at the student level (4 of 7). Although students felt teachers' explanations sometimes did not speak to them, they still perceived them as valuable support when the teachers devoted sufficient time and gave students room to shape the explanations to their needs. Four students (P01, P03, P05, P06) depicted that on-demand one-on-one tutoring with teachers was the ideal support format when they had programming problems. As P05 described, one ideal support would be:

> *"One-on-one with a teacher and he could explain to me in words I could understand ... And if I don't understand the way he explains it to me, I could like ask him to give me a metaphor or like some images or compared to something in that but is in everyday life."* (P05)

P01 echoed with this idea in her ideal support design as *"being able to be on one-on-one with the teacher ... learning like one-on-one, it's easier to understand because they're teaching it in the way that you know the best."* She also described a previous supportive experience as when the teacher was able to *"explain it to me in an easier way, use easier words in terms. Because when I learned, it's easier for me to understand things when it's in less words."*

Enough guidance but encourage independent thinking (4 of 7). Even when minoritized women needed support to make progress, they still emphasized the importance of maintaining independent thinking and developing the ability to solve programming tasks without help (P02, P03, P09, P11). P03 characterized the positive outcome of her ideal support as *"I finish the lab (homework assignments) using collaboration and my own thoughts."* P11 expected the outcome to be *"students will know exactly where they made a mistake and how to fix it instinctively in the future, instead of just memorizing an answer key."* To achieve this, she presented a sophisticated support design to balance AI help and students' self-contribution:

> *"Offer students the ability to check their work with AI-generated code at the end of an assignment and explanations of both the students' errors and common errors. To encourage students to try the work themselves before turning to the available help, here would be a limit on how often the service can be used, and teachers can choose to block it for graded assignments."*

4.2.2 Code examples as support (6 of 12): Context-specific, Variation-covered, Knowledge- and generation-fit. When struggling, students obtain code examples to clarify the problem and transfer the takeaways to the tasks they are working on. In current practices, they either actively collect or receive those examples from different sources, including instructors (P07, P11), official documentations (P02), textbooks (e.g., online E-books) (P08, P11), and other online tutorials (e.g., Google, TikTok, Khan Academy, YouTube, Code.org) (P04, P05, P08). Students reported that, while examples might be widely available, only a subset provided meaningful support.

Context-specific and cover various use cases (4 of 6). Students tried to get more context-targeted examples because more universal examples could be hard to apply to their specific problems. For example, P08 envisioned an ideal support as *"the examples of how to do it before the assignment .. not an exact solution to the*

assignment, but a similar problem', and explained its helpfulness as *"because then I wouldn't be lost during the assignment ... I know how to implement that into the code."* (P08) Similarly, P11 also described how basic examples helped her get started, but more targeted examples were needed to progress further, *"I could only complete the basic code. I could not expand my ideas to create my own ... designs, I had to more or less copy the example."* P02 wanted more interactive, video-based examples and emphasized the importance of explaining how to use specific methods. As an ideal support, P07 also emphasized the need for broader examples to accommodate the varying types of uses for a single programming concept. She detailed it as: *"as many different situations that's possible ... how would you access each of those if they were like different ways of accessing them."*

Knowledge-level and generationally relevant (2 of 6). An effective code example needs to match students' current knowledge level. When P04 envisioned her ideal support, she noted that the instructor often picked complex examples that only increased the challenge of grasping difficult concepts. Therefore, she wanted to use AI to *"provide more examples of problems we actually go over in class and simple it down more."* (P04). When it comes to AI inaccuracies, she relied on human checks to get rid of them, *"make sure the coding examples are verified by real people"* (P04). Other students also shared searching tricks in their current practices to find examples that were appropriate to their knowledge level: *"if you search up with 'AP' (as the keywords) ... that's when it's going to start giving you something that's more like specific to what you want."* (P09) In addition, in P04's design, she wanted the teachers to make materials that felt more relevant and generationally connected, helping them focus better:

> *"The teachers should have better resources and something that'll fit my generation because we lose focus and get bored easily, so something that's, I can't explain it, but if it appeals to like our generation we're gonna be more bound to like pay attention to it ... do more examples on things based on our real lives."* (P04)

4.2.3 Diagnostic information as support (4 of 12): Timely, Actionable and Habit-fostered. In addition to information that was more explanatory and illustrative, students (P03, P10, P11, P12) emphasized the need for precise and detailed diagnostic information to reveal their misconceptions and skill gaps clearly.

Highlight what works, what does not, and how to fix it (3 of 4). This included an immediate snapshot of their current progress (P10, P12), highlighting both what was working well and what was not. Depending on their current progress in problem-solving, students wanted different areas of focus. More directional information ("what to do") was preferred at the beginning, while more reflective information ("why it is here") was valued in later stages (P03, P10). The ideal support should also provide precise error locations and actionable suggestions for correction (P03, P11, P12):

> *"An AI that is not based in machine learning and had preconditioned functions and information. It would be a support button walking through and explaining a program when needed, and where you went wrong/ what to do and why.* (P12)

Not just correctness, but better coding habits (1 of 4). Interestingly, P10 is a minoritized woman who holds multi-faceted motivation to learn programming, which includes personal passion in technology, future career goals in technology or computing, finding it fun and interesting, and the chance to earn college credit early through it. Beyond correctness, in her design, she also sought to refine her coding habit to produce cleaner code.

> *"My idea is that like I put my code into it and it tells me what's wrong with it ... how to correct my coding habits and make it look better and make it easier to understand. It's like not just if it's not working, but also what habits and what steps I should take to make it cleaner."* (P10)

4.3 RQ3: What *matters* to minoritized women students when determining desired support in programming learning?

In this section, we report students' perceptions of possible programming support resources, including their preferences, benefits, and concerns. They went through candidate programming support resources, explained which to keep or remove, ranked them, and discussed how these could impact programming problem-solving practices. Although students might vary in which features they keep or remove, their underlying values and concerns were shared.

4.3.1 What benefits make a support resource valuable? Minoritized women students tended to value a support resource if it provided a visible contribution to solving their struggles, offered thoughtful encouragement, made connections with trusted help providers, promoted self-reflection and self-review, and helped develop broader learning strategies.

Visible contributions in solving the issue. A valuable programming support, regardless of type, should clearly aid issue resolution. This was a key criterion that students mentioned during the candidate selection process across all the students. However, this perceived contribution differed by feature across individuals. For example, two students (P05 and P12) found the feature "a code example from a similar problem" not valuable because *"don't know why that code example, like why does it relate to what I'm working on."* (P05) But other students valued this because they could sense a clear contribution: *"it can help you understand the general vibe of the code"* (P11) and *"it helps you look through it and find out like what you're missing in your code and why they use certain things."* (P09).

Develop broader strategies beyond a single problem. While students focused on a specific problem, some went beyond it and appreciated the chance to acquire broader learning strategies. Ten minoritized women kept the resources that allowed them to pause a problem and either return later, stop, or switch tasks. This could be a reminder to prevent brain overload, *"I feel that's very helpful for the coding process, because you might not see something when you're looking at it very intensely."* (P06) An external reminder could be more accountable and harder to dismiss, *"Sitting down for so long is not really good for you. So just knowing that, hey, you can like take a break, come back to it when your mind is more relaxed would help."* (P03) Additionally, it could contribute to better planning and

prioritization skills, *"I'm the kind of person who can get stuck on the same problem for a long time, so it's good for pacing and time management."* (P11) Six students also kept the feature "a letter from a previous student describing their learning experience", which not only motivated them but also gave them strategies to overcome similar challenges.

Provide encouragement in a thoughtful way. Because learning programming could be discouraging for students, nine students chose to keep resources that made them feel encouraged. In particular, 7 students wanted to keep the feature "encouraging messages to continue". P04 explained the reason as *"because I feel like a lot of computer science kids get discouraged very quickly"*. P05 shared a similar sense, *"because sometimes it can feel like a little bit frustrating when I don't immediately understand what's going on."* However, students did not want blind encouragement, as P08 pointed out, *"but it wouldn't be efficient to keep going if you're just like completely lost because you're probably not going to get there."* And such encouragement was more valuable when grounded in empirical, real-world pathways: *"Well, if I know that someone else had the same struggle but pushed through and was successful, then maybe I can be too."* (P02)

Provide a comfortable connection to trusted helpers. Students held different levels of trust in potential support sources, and they retained features that link them to reliable ones. For example, P10 held a lack of faith in support from teachers or peers, *"I've taken four online classes regarding computing. Whenever I message the teacher, they just give me the answer. So they aren't really helping me ... haven't really found much help from friends and classmates. They usually aren't any help. I won't lie."* Therefore, she only kept AI, *"I think that chatbots are extremely helpful in learning how to code. Like, I'm pretty sure every single person I know that has learned how to code from their own home. ChatGPT or Co-pilot has been a part of their journey."*, and online forums, *"Discuss about the problem with strangers on the internet. Love that. I think strangers on the internet can be very helpful when safely utilized."* Students also valued anonymity as a more comfortable way to ask for help in known connections (8 of 12), particularly when they already felt minoritized in a classroom: *"It'd be easier not to have that sort of discrimination against you. So I think just being able to discuss with people about computing problems and you can share difficulties without the fear of getting what discriminated against based on certain aspects about you that you can't control."* (P03)

Promote self-reflection and self-review. Students also valued the feature that could nudge them to do self-reflection and self-assessment, addressing gaps through review if needed (4 of 12). P11 preferred the feature showing "information on how many other students are stuck at the same point before", because *"if you see the only two other students are stuck with this, then it might mean you used to review an earlier concept. But if everybody's struggling with it, you know it's just a difficult concept."* P04 also kept the feature "information on how many other students have completed this problem" due to a similar reason, *"it can help me know like what I need to catch up and what I need to work on myself."*

4.3.2 What concerns might make a support resource less valuable? Students found resources less valuable if perceived as unnecessary, irrelevant, or insufficient; if they caused negative impacts on social dynamics or a possible loss of autonomy; or if they were linked to sources distrusted due to prior negative experiences.

Unnecessary and irrelevant to the issue. Students considered a feature unvaluable if it seemed unnecessary or irrelevant to their struggles. Seven students did not like being given a simpler problem on the topic when they were struggling, *"Sometimes it's the harder part that you're confused about, so an easier problem might not necessarily help you in that situation."* (P09). While additional test cases served to clarify requirements and edge cases for some students, others did not see it as necessary: *"that doesn't necessarily help. If your code is wrong, giving more input and output examples will just show you more that your code is wrong, but it won't necessarily help creating a solution."*

Relatedly, information on how many others got stuck at that point before could normalize some students' struggles and prompt self-reflection. However, this could be unhelpful for others since it *"doesn't necessarily actually help you with your code."* (P09) Some students perceived little relevance from information about others in general (3 of 12), as their primary focus was on resolving their own difficulties, *"knowing the information about other students doesn't help me at all."* (P02)

Negative impact on classroom dynamics and affective state. Students raised concerns that some resources might cause stress. For example, advising a time limit on problem-solving can add pressure, *"I feel like it would kind of stress me out."* (P04). Additionally, viewing progress information from peers, whether it is their successes or struggles, can lead to unnecessary competition (11 students removed at least one of the related feature candidates). Students worried that such information could lead to *"a contest that's not really needed"* (P01) and even negative self-assessment, *"it just makes you feel more pressured and feel like you're not doing it as fast as fast as you should be."* (P09) In addition, they noted that the number of students shown in such information could be demotivating: *"In some cases, it can be very discouraging if it's considered a relatively easy problem, but the person struggling on it, and not other people have struggled on it, it could be very discouraging."* (P12)

A possible loss of autonomy. Some students worried that excessive suggestions from other resources might take away their sense of control over problem-solving. P10 disliked the time suggestion feature since *"I just don't think that they should be telling me how much time to spend."* (P10). This concern also led her to remove a feature about the needed concepts to create a solution, *"I think it would make me overthink when I'm planning out my code."* (P10) P04 also felt that more detailed error messages could mislead her attention, *"I feel like we all know why it's wrong already, but we need to know how to fix it. So I feel like we shouldn't prioritize their message."* (P04)

Four students were concerned that receiving letters from previous students ignored their own uniqueness, *"it wouldn't be helpful because you wouldn't necessarily know or you wouldn't be in the same position as a student, even if you're doing super well. Because everyone's pathway is very different."* (P12)

Insufficient and partial assistance. Students reported that certain resources provided only partial, middle-stage support, which was insufficient. For example, having all the programming concepts required for a solution could act as a helpful checklist or starting point for some (7 of 12), but it is not enough for others because *"even*

though you know the concepts, it might not really give you how to put it together in a way that you need to actually create the solution." (P09).

Likewise, more detailed error messages were inadequate for some students who wanted guidance on *how* rather than *what*, *"I have used some programs where it has told me exactly where I'm wrong, but not exactly how to fix it or I could fix it."* (P12) The feature showing how accurate a solution was received comparable critique, *"if only knew how accurate it was, I wouldn't know how to get better unless I was given how. Because if I just knew how accurate it was, I wouldn't be able to do anything else."* (P07)

Distrust of quality due to prior poor experience. Sometimes, students decided against a potential feature because of prior negative experience. One resource was linking students with corresponding textbook materials. Five students rejected it due to unhappy personal experiences, finding it wordy and passive: *"Some people don't want to read. It's so tedious, and it's a lot."* (P02) and *"They're very wordy, or it's just a lot to sift through. And it doesn't make me motivated to actually try to find it because it's just like a lot of words."* (P09)

Correspondingly, although web-search could provide valuable resources, five students chose not to keep it because capturing effective content from Internet searches could be time-consuming and ineffective: *"I think that the internet is really big, so it wouldn't be very helpful in narrowing down the problem"* (P06) and *"Looking up how to code on Google is so ineffective... All these sources come up, you have to read through these extremely lengthy texts and articles."* (P10)

5 Discussion

Above, we first summarized the challenges that minoritized women students face when seeking support while learning programming, such as resource shortages, perceived pressure, impractical content, and the tension between speed and accuracy (Section 4.1). Then we identified key design dimensions from their experiences and designs for desired programming learning support, including explanatory information, illustrative examples, and diagnostic information with specific characteristics (Section 4.2). Finally, we discussed students' perspectives on a broader range of potential support resources, highlighting the benefits that make it desirable and the concerns that reduce its value (Section 4.3). In this discussion, we first interpret our findings through an asset-based lens [2, 83, 91, 130], connect them with established learning science insights and research on minoritized students' help-seeking in computing learning (see subsection 5.1). Then, we connect students' designs, values, and concerns to articulate several dual-facet design goals and implications for next-generation support systems, better equipping students in computing education (see subsection 5.2).

5.1 Students' experience-rooted designs for desired programming learning support

Although we did not initially adopt an asset-based design framework, it emerged as a useful lens for interpreting the minoritized women students' self-designs [2, 83, 130]. When depicting their ideal programming learning supports, students in our study did not aim to invent novel technologies [83]. Instead, they focused on adapting existing supports to provide more desired support and amplifying those that already present [83].

Specifically, many of their proposed designs (see subsection 4.2) were built on common instructional content types, such as examples, explanations, and feedback [1, 18, 115]. They then built on their own learning experiences to describe specific desired characteristics. These students did not intentionally draw on learning science principles in their designs, yet many of their ideas naturally align with them. For example, the design of explanatory information with a peer-level tone and a calibrated expert perspective, and the context-specific, knowledge-appropriate, and generationally relevant code examples align well with efforts to reduce extraneous cognitive load [114]. Additionally, the idea of providing enough guidance while still encouraging independent thinking aligns with the Zone of Proximal Development [121], which keeps students working at an optimal level of challenge, and the idea of productive struggle [44]. Moreover, the design of diagnostic information is consistent with the recommended design of elaborated instructional feedback (e.g., response contingent, topic contingent, and misconceptions) [110] and the efforts to improve error messages in text-based programming learning [8]. These also reflect a meaningful interaction between participatory design and the learning sciences [28].

The designs proposed by students also resonated with previous research on minorities within computing education. For instance, most students (11 of 12) continued to deeply value teachers' support and intentionally designed features that leveraged it, such as structured support sequences that begin with peers and progress to teachers, or increased opportunities for one-on-one sessions. This aligns with prior findings indicating that greater collaboration with teachers is associated with stronger intentions for minoritized women to persist in computing [33], and that women use formal [29] and internal resources [54] more frequently. In addition, 8 of 12 women favored the anonymous forum, which aligned with the finding that women preferred to post anonymously in forums in computing classes [42, 61]. Moreover, 11 of 12 women removed at least one feature about knowing peers' progress, aligning with research showing that concerns of social comparison weigh more heavily on women in STEM [58]. Their designs to seek social support are also found to be a useful factor that supports minoritized women's persistence in computing learning [36].

For the relationship with new technology, students in our study recognized emerging technologies and platforms, such as AI and TikTok, for learning. Some had already incorporated these into their everyday programming learning. However, since school remains the primary environment where most of them spend their day, many potential help providers they identified still came from everyday roles around them, such as teachers, friends, and classmates. Some of their support designs also adapted existing infrastructures in other domains to programming learning contexts, such as after-class tutoring sessions.

Prior research has shown that, to build engagement and retention among minoritized groups in computing, sustained efforts are necessary to transform the learning journeys of minoritized computing learners over time [37, 100]. Therefore, with consideration for sustaining effective support design over the long term [47], education technology designers should focus on enhancing what

already exists, but be cautious about introducing additive elements that might further expand the information scope and effort required to obtain valuable help [130]. For example, designers can adopt modular designs that integrate into students' existing learning flows, rather than creating and adding entirely new, hard-to-maintain software or content types [126]. Additionally, while new technologies like the Internet and AI enable more technology-mediated support, many students still design around human social support systems, rather than relying solely on technology. When technology appears capable of handling more programming help requests [38], it is more important than ever to emphasize the value of human-to-human interaction and to design better social support systems in programming learning [38, 81], especially among minority learner groups [97, 98, 116].

5.2 Twofold design needs and implications for programming learning support

Our findings highlight several shared expectations about desired programming support for minoritized women students. These expectations can appear contradictory, but are collectively important. We summarize four key themes and integrate insights from recent education tool designs to recommend ways to meet these jointly held needs.

5.2.1 Offer problem-specific help with bigger-picture covered. Our findings highlight a key pair of goals underlying students' desired programming support. Help-seeking is a meta-cognitive process tied to self-regulated learning [41, 45, 67, 87]. A key step during this process is deciding on the help-seeking goals. Some students focused on solving the immediate issue in a given problem, and therefore only valued the support that could directly help them progress. Other students not only appreciated resources that addressed the immediate issue, but also valued those that revealed their knowledge gaps and offered opportunities for review. This suggests an expansion of help-seeking goals, extending from the specific issue at the core to underlying areas of missing knowledge. Therefore, future programming support systems should articulate the levels of support they offer and transition from task-specific guidance to broader conceptual understanding.

5.2.2 Deliver thorough guidance while maintaining independent thinking. A taxonomy categorizes help types into instrumental help, which focuses on the new skill-building process, and executive help, which emphasizes the outcome while omitting the process [34, 71]. Students in our study generally underlined the importance of preserving their own thinking with suitable instrumental help. Therefore, while many appreciated being shown a solution or using AI as a resource, they often put it as support to be used only under specific conditions, such as a last resort. In addition, learning from failures and struggles is an essential aspect of self-growth [56, 122] and important in computing education [43, 90]. However, a key to helping students learn from failures is to decrease their fear of failure [70, 90], as women often fear failure more in computing and engineering, which may hinder persistence [25, 75, 117]. In this case, it is recommended that future programming learning systems implement stepped support [50] and inform students of the availability of more extensive help after sustained effort, thereby reducing mental burden and fostering fearless learning.

5.2.3 Foster social belonging without triggering pressure. Our findings indicate a need for reassurance of not struggling alone, without the stress of comparison. This predominantly arises when it comes to knowing information about others, which is a common challenge in social interactions and personal informatics [68, 120]. Our findings suggest that, in the context of designing programming learning support, the choice of information metrics [120] is critical for conveying a sense of social belonging, social presence [127], and facilitating social help resources [52] rather than competitive comparison. Future research should identify which learning metrics feel sensitive to share with peers and which can be used for peer-support grouping without judgment. What's more, this study examined only a snapshot of computing learning. To fully understand students' needs for social support or role models in different learning stages, broader longitudinal evaluations covering diverse education decision milestones (e.g., first course, college board courses, computing majors) are needed.

5.2.4 Keep student agency in help-seeking with minimal risk. Another key point is who to initiate the help-seeking. While traditional help-seeking is usually initiated by humans themselves [48], many students, especially minority learners [129], struggle with the fear of help rejection or help-seeking threat [77]. Minoritized women in our study reported similar discomfort, which at times led them to avoid seeking help after experiencing learning challenges. The advances in technology provide an opportunity for more proactive support initiated by the system [18, 66, 87, 113, 128]. However, most of these are focused on system-initiated and system-generated support. Given that many students still place a high value on help from teachers, it is important to provide not only system support but also system-initiated, teacher-led support. Moreover, as help-seeking is a valuable meta-cognitive skill to develop [40, 59], proactive assistance is expected to fade as students become more comfortable seeking help from others and reclaiming their own agency.

6 Limitations & Future Work

This work has several limitations. First, because the study centers on designing with students, the ideas and designs originated from student self-reporting. Given that students' perspectives are often grounded in their own lived experiences, they may not generalize across contexts or align with existing learning principles. Second, these design ideas still lack empirical validation. Future empirical evaluations are required to determine whether these ideas actually support learning. Third, the session format was chosen to allow for deeper and more contextualized insights into each participant's perspective. Future work could explore collaborative sessions to understand how students expand on and refine each other's design ideas. Fourth, all the students in this work were already enrolled in AP CS courses and interested in participating in our study. Given that many under-resourced schools did not even offer AP CS courses, this introduced selection bias based on participants' motivation and confidence in succeeding in computing, which may have limited the diversity and complexity of help-seeking and support practices. Future studies could examine students who chose to

leave computing education before AP CS or who never considered opting in. Finally, participants in our study were recruited through two specific channels within one country. Cultural and regional differences in computing education might impact the applicability of the results to other areas and learner groups. Future research should broaden its scope by including students from diverse countries and various learning stages. We also acknowledge that the small sample size limits the generalizability and representativeness of our findings.

7 Conclusion

Although computing has gained increasing attention in K–12 education, women of minoritized racial and ethnic groups remain underrepresented. We conducted design sessions with 12 minoritized high school women to understand their help-seeking challenges and the desired support in formal programming classes. Four key challenges emerged in current practices: the poor-quality resources (limited, timely yet inaccurate, valid yet difficult to apply), discomfort in seeking human help, targeted strategies for getting help from teachers, and unclear instructions that undermined other support. From minoritized women's designs, we identified three key content types and their desired characteristics: (1) explanatory information (expert-level credibility, peer-level tone, space for thinking); (2) code examples (context-specific, variation-covered, knowledge- and generation-fit); and (3) diagnostic information (timely, actionable, and habit-fostering). Additionally, we found that students valued resources that demonstrate a clear link to their issues, offer thoughtful encouragement, advance self-review and broader learning strategies, and connect them with trusted helpers. In contrast, they devalued resources that were unnecessary, irrelevant, or insufficient, negatively affected social dynamics, reduced their autonomy, or were from unreliable sources. This work presents valuable design implications derived from the experiences and perspectives of underrepresented students in computing learning, which can inform future educational design and technology.

Acknowledgments

This work was supported by NSF award #2143028. Any conclusions expressed in this material do not necessarily reflect the views of NSF.

References

[1] Mohammad Abolnejadian, Sharareh Alipour, and Kamyar Taeb. 2024. Leveraging chatgpt for adaptive learning through personalized prompt-based instruction: A cs1 education case study. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. 1–8.

[2] Veronica Ahumada-Newhart, J Maya Hernandez, and Karla Badillo-Urquiola. 2021. A call for action: Conceptualizing assets-based inclusive design as a social movement to address systemic inequities: An assets-based inclusive design framework. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–4.

[3] Vincent Aleven, Elmar Stahl, Silke Schworm, Frank Fischer, and Raven Wallace. 2003. Help seeking and help design in interactive learning environments. *Review of educational research* 73, 3 (2003), 277–320.

[4] Jamie Amemiya and Ming-Te Wang. 2017. Transactional relations between motivational beliefs and help seeking from teachers and peers across adolescence. *Journal of youth and adolescence* 46, 8 (2017), 1743–1757.

[5] Nisha Anthraper, Prachee Javiya, Sai Iluru, Lujie Karen Chen, and Andrea Kleinsmith. 2024. PeerConnect: Co-Designing a Peer-Mentoring Support System with Computing Transfer Students. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. 1–7.

[6] Monica Babes-Vroman, Isabel Juniewicz, Bruno Lucarelli, Nicole Fox, Thu Nguyen, Andrew Tjang, Georgiana Haldeman, Ashni Mehta, and Risham Chokshi. 2017. Exploring gender diversity in CS at a large public R1 research university. In *Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education*. 51–56.

[7] Katherine Bartlett. 2018. *Feminist legal theory: Readings in law and gender*. Routledge.

[8] Brett A Becker, Graham Glanville, Ricardo Iwashima, Claire McDonnell, Kyle Goslin, and Catherine Mooney. 2016. Effective compiler error message enhancement for novice programming students. *Computer Science Education* 26, 2-3 (2016), 148–175.

[9] Phillip A Boda and Steven McGee. 2021. Broadening participation and success in AP CSA: Predictive modeling from three years of data. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. 626–632.

[10] Jürgen Börstler, Mark S Hall, Marie Nordström, James H Paterson, Kate Sanders, Carsten Schulte, and Lynda Thomas. 2010. An evaluation of object oriented example programs in introductory programming textbooks. *ACM SIGCSE Bulletin* 41, 4 (2010), 126–143.

[11] Virginia Braun and Victoria Clarke. 2019. Reflecting on reflexive thematic analysis. *Qualitative research in sport, exercise and health* 11, 4 (2019), 589–597.

[12] Virginia Braun and Victoria Clarke. 2021. Thematic analysis: A practical guide. (2021).

[13] Virginia Braun and Victoria Clarke. 2023. Toward good practice in thematic analysis: Avoiding common problems and be (com) ing a knowing researcher. *International journal of transgender health* 24, 1 (2023), 1–6.

[14] Cara Broadley. 2020. Advancing asset-based practice: Engagement, ownership, and outcomes in participatory design. *The Design Journal* 24, 2 (2020), 253–275.

[15] Paul Bruno, Mariam Saffar Pérez, and Colleen M Lewis. 2022. Four Practical Challenges for High School Computer Science. *Policy Analysis for California Education, PACE* (2022).

[16] Kelly Caine. 2016. Local standards for sample size at CHI. In *Proceedings of the 2016 CHI conference on human factors in computing systems*. 981–992.

[17] Liuqing Chen, Shuhong Xiao, Yunnong Chen, Yaxuan Song, Ruoyu Wu, and Lingyun Sun. 2024. ChatScratch: An AI-augmented system toward autonomous visual programming learning for children aged 6-12. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–19.

[18] Valerie Chen, Alan Zhu, Sebastian Zhao, Hussein Mozannar, David Sontag, and Ameet Talwalkar. 2025. Need help? designing proactive ai assistants for programming. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–18.

[19] Yuk Fai Cheong, Frank Pajares, and Paul S Oberman. 2004. Motivation and academic help-seeking in high school computer science. *Computer Science Education* 14, 1 (2004), 3–19.

[20] Alexander Cho, Roxana G Herrera, Luis Chaidez, and Adilene Uriostegui. 2019. The" Comadre" project: an asset-based design approach to connecting low-income Latinx families to out-of-school learning opportunities. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–14.

[21] Code.org. 2026. *Computer Science Access and Participation Data*. https://advocacy.code.org/report-data/

[22] Code.org. 2026. *Dig Deeper into AP Computer Science*. https://code.org/en-US/computer-science-advanced-placement-data

[23] Kathryn Cunningham, Barbara J Ericson, Rahul Agrawal Bejarano, and Mark Guzdial. 2021. Avoiding the Turing tarpit: Learning conversational programming by starting from code's purpose. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–15.

[24] Scott Davidoff, Min Kyung Lee, Anind K Dey, and John Zimmerman. 2007. Rapidly exploring application design through speed dating. In *International conference on ubiquitous computing*. Springer, 429–446.

[25] Mary E Dawes, John J Horan, and Gail Hackett. 2000. Experimental evaluation of self-efficacy treatment on technical/scientific career outcomes. *British Journal of Guidance & Counselling* 28, 1 (2000), 87–99.

[26] Linda DeAngelo, Danielle Vegas Lewis, and Morgen Snowadzky. 2025. Caught in a double bind: Toward an understanding of undergraduate women engineers early experiences. *The Review of Higher Education* (2025).

[27] Paul Denny, Stephen MacNeil, Jaromir Savelka, Leo Porter, and Andrew Luxton-Reilly. 2024. Desirable characteristics for AI teaching assistants in programming education. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*. 408–414.

[28] Betsy DiSalvo, Jason Yip, Elizabeth Bonsignore, and DiSalvo Carl. 2017. Participatory design for learning. In *Participatory design for learning*. Routledge, 3–6.

[29] Augie Doebling and Ayaan M Kazerouni. 2021. Patterns of academic help-seeking in undergraduate computing students. In *Proceedings of the 21st Koli Calling International Conference on Computing Education Research*. 1–10.

[30] Sheena Erete, Yolanda Rankin, and Jakita Thomas. 2023. A method to the madness: Applying an intersectional analysis of structural oppression and power in HCI and design. *ACM Transactions on Computer-Human Interaction* 30, 2 (2023), 1–45.

[31] Danyang Fan, Olivia Tomassetti, Aya Mouallem, Gene SH Kim, Shloke Nirav Patel, Saehui Hwang, Patricia Leader, Danielle Sugrue, Tristen Chen, Darren Reese Ou, et al. 2025. Promoting Comprehension and Engagement in Introductory Data and Statistics for Blind and Low-Vision Students: A Co-Design Study. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–20.

[32] Rachel Figard, Sri Yash Tadimalla, and Emma R Dodoo. 2023. More than a checkbox: Exploring intersectional experiences of engineering students using the Social Identity Wheel. In *2023 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–5.

[33] Susan R Fisk, Brittany Watts, Courtney Dress, Charlotte Lee, Audrey Rorrer, Tom McKlin, Tiffany Barnes, and Jamie Payton. 2024. Retaining Black women in computing: A comparative analysis of interventions for computing persistence. *ACM Transactions on Computing Education* 24, 2 (2024), 1–25.

[34] Carlton J Fong, Cassandra Gonzales, Christie Hill-Troglin Cox, and Holly B Shinn. 2023. Academic help-seeking and achievement of postsecondary students: A meta-analytic investigation. *Journal of Educational Psychology* 115, 1 (2023), 1.

[35] Zhikai Gao, Adam Gaweda, Collin Lynch, Sarah Heckman, Damilola Babalola, and Gabriel Silva de Oliveira. 2024. Using Survival Analysis to Model Students' Patience in Online Office Hour Queues. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2*. 1646–1647.

[36] Xingjian Gu and Barbara J Ericson. 2025. The Intersectional Experience of Black Girl High School Students in Advanced Placement Computer Science. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*. 402–408.

[37] Mark Guzdial, Barbara Ericson, Tom Mcklin, and Shelly Engelman. 2014. Georgia computes! An intervention in a US state, with formal and informal education in a policy context. *ACM Transactions on Computing Education (TOCE)* 14, 2 (2014), 1–29.

[38] Irene Hou, Owen Man, Kate Hamilton, Srishty Muthusekaran, Jeffin Johnykutty, Leili Zadeh, and Stephen MacNeil. 2025. 'All Roads Lead to ChatGPT': How Generative AI is Eroding Social Interactions and Student Learning Communities. In *Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1*. 79–85.

[39] Irene Hou, Sophia Mettille, Owen Man, Zhuo Li, Cynthia Zastudil, and Stephen MacNeil. 2024. The effects of generative AI on computing students' help-seeking preferences. In *Proceedings of the 26th australasian computing education conference*. 39–48.

[40] Nathaniel Hudson, Benjamin Lafreniere, Parmit K Chilana, and Tovi Grossman. 2018. Investigating how online help and learning resources support children's use of 3D design software. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–14.

[41] Nikhita Joshi, Justin Matejka, Fraser Anderson, Tovi Grossman, and George Fitzmaurice. 2020. Micromentor: Peer-to-peer software help sessions in three minutes or less. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.

[42] David A Joyner, Lily Bernstein, Ian Bolger, Maria-Isabelle Dittamo, Stephanie Gorham, and Rachel Hudson. 2022. Anonymity: A double-edged sword for gender equity in a cs1 forum?. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education-Volume 1*. 766–772.

[43] Maria Kallia. 2025. To Be, or to Be Otherwise? Silicon Souls in Search of Dasein and Authentic Human Engagement in the Age of Generative AI. In *Proceedings of the 2025 ACM Conference on International Computing Education Research V. 1*. 240–255.

[44] Manu Kapur. 2008. Productive failure. *Cognition and instruction* 26, 3 (2008), 379–424.

[45] Stuart A Karabenick and Jean-Louis Berger. 2013. Help seeking as a self-regulated learning strategy. *Applications of self-regulated learning across diverse disciplines: A tribute to Barry J. Zimmerman* (2013), 237–261.

[46] Stuart A Karabenick and Myron H Dembo. 2011. Understanding and facilitating self-regulated help seeking. *New directions for teaching and learning* 2011, 126 (2011), 33–43.

[47] Finn Kensing and Jeanette Blomberg. 1998. Participatory design: Issues and concerns. *Computer supported cooperative work (CSCW)* 7, 3 (1998), 167–185.

[48] Kimia Kiani, George Cui, Andrea Bunt, Joanna McGrenere, and Parmit K Chilana. 2019. Beyond" One-Size-Fits-All" Understanding the Diversity in How Software Newcomers Discover and Make Use of Help Resources. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–14.

[49] Richard Kick and Frances P Trees. 2015. AP CS principles: engaging, challenging, and rewarding. *ACM Inroads* 6, 1 (2015), 42–45.

[50] Minsol Kim, Aliea L Nallbani, and Abby Rayne Stovall. 2024. Exploring LLM-based Chatbot for Language Learning and Cultivation of Growth Mindset. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. 1–5.

[51] Shao-Heng Ko and Kristin Stephens-Martinez. 2023. What drives students to office hours: individual differences and similarities. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. 959–965.

[52] Shao-Heng Ko and Kristin Stephens-Martinez. 2024. The Trees in the Forest: Characterizing Computing Students' Individual Help-Seeking Approaches. In *Proceedings of the 2024 ACM Conference on International Computing Education Research-Volume 1*. 343–358.

[53] Shao-Heng Ko and Kristin Stephens-Martinez. 2025. Prior What Experience? The Relationship Between Prior Experience and Student Help-Seeking Beyond CS1. In *Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1*. 100–106.

[54] Shao-Heng Ko, Matthew Zahn, Kristin Stephens-Martinez, Yesenia Velasco, Lina Battestilli, and Sarah Heckman. 2025. Relationships Between Computing Students' Characteristics, Help-Seeking Approaches, and Help-Seeking Behavior in Introductory Courses and Beyond. In *Proceedings of the 2025 ACM Conference on International Computing Education Research V. 1*. 313–326.

[55] Sonia Koshy, Alexis Martin, Laura Hinton, Allison Scott, Bryan Twarek, and Kalisha Davis. 2021. The computer science teacher landscape: Results of a nationwide teacher survey. *Kapor Center* (2021).

[56] Julius Kuhl, Markus Quirin, and Sander L Koole. 2021. The functional architecture of human motivation: Personality systems interactions theory. In *Advances in motivation science*. Vol. 8. Elsevier, 1–62.

[57] Sari Kujala. 2003. User involvement: a review of the benefits and challenges. *Behaviour & information technology* 22, 1 (2003), 1–16.

[58] Hyewon Lee, L Yu Shirley, Minjung Kim, and Alison C Koenka. 2021. Concern or comfort with social comparisons matter in undergraduate physics courses: Joint consideration of situated expectancy-value theory, mindsets, and gender. *Contemporary Educational Psychology* 67 (2021), 102023.

[59] HaeJin Lee, Frank Stinar, Ruohan Zong, Hannah Valdiviejas, Dong Wang, and Nigel Bosch. 2025. Learning Behaviors Mediate the Effect of AI-powered Support for Metacognitive Calibration on Learning Outcomes. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–18.

[60] Irene A Lee, Maureen Psaila Dombrowski, and Ed Angel. 2017. Preparing stem teachers to offer new mexico computer science for all. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. 363–368.

[61] Ryan Lenfant, Alice Wanner, John R Hott, and Raymond Pettit. 2023. Project-based and assignment-based courses: A study of piazza engagement and gender in online courses. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*. 138–144.

[62] Dan Leyzberg and Christopher Moretti. 2017. Teaching CS to CS teachers: Addressing the need for advanced content in K-12 professional development. In *Proceedings of the 2017 ACM SIGCSE technical symposium on Computer Science Education*. 369–374.

[63] Chun Li, Jaemarie Solyst, Safiyyah Scott, Gabriella Howse, Tara Nkrumah, Erin Walker, Amy Ogan, and Angela EB Stewart. 2025. " I am a Technology Creator": Black Girls as Technosocial Change Agents in a Culturally-Responsive Robotics Camp. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–14.

[64] Ally Limke, Marnie Hill, Veronica Cateté, and Tiffany Barnes. 2024. A survey of K-12 teacher needs for an online programming learning system. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. 1–7.

[65] Ally Limke, Saminur Islam, Bahare Riahi, Xiaoyi Tian, Marnie Hill, Veronica Cateté, and Tiffany Barnes. 2025. What Does It Take to Support Problem Solving in Programming Classrooms? A New Framework from the K-12 Teacher Perspective. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. 1–7.

[66] Xingyu Bruce Liu, Shitao Fang, Weiyan Shi, Chien-Sheng Wu, Takeo Igarashi, and Xiang 'Anthony' Chen. 2025. Proactive conversational agents with inner thoughts. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–19.

[67] Dastyni Loksa, Lauren Margulieux, Brett A Becker, Michelle Craig, Paul Denny, Raymond Pettit, and James Prather. 2022. Metacognition and self-regulation in programming education: Theories and exemplars of use. *ACM Transactions on Computing Education (TOCE)* 22, 4 (2022), 1–31.

[68] Blake M. DiCosola III and Gina Neff. 2022. Nudging behavior change: using in-group and out-group social comparisons to encourage healthier choices. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–14.

[69] Lena Mamykina, Bella Manoim, Manas Mittal, George Hripcsak, and Björn Hartmann. 2011. Design lessons from the fastest q&a site in the west. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 2857–2866.

[70] Lauren E Margulieux, James Prather, Brent N Reeves, Brett A Becker, Gozde Cetin Uzun, Dastyni Loksa, Juho Leinonen, and Paul Denny. 2024. Self-regulation, self-efficacy, and fear of failure interactions with how novices use llms to solve programming problems. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*. 276–282.

[71] Sergi Martin-Arbos, Elena Castarlenas, and Jorge-Manuel Duenas. 2021. Help-seeking in an academic context: A systematic review. *Sustainability* 13, 8 (2021), 4460.

[72] Samiha Marwan, Bita Akram, Tiffany Barnes, and Thomas W Price. 2022. Adaptive immediate feedback for block-based programming: Design and evaluation. *IEEE Transactions on Learning Technologies* 15, 3 (2022), 406–420.

[73] Samiha Marwan, Preya Shabrina, Alex Milliken, Ian Menezes, Veronica Catete, Thomas W Price, and Tiffany Barnes. 2021. Promoting students' progress-monitoring behavior during block-based programming. In *Proceedings of the 21st Koli Calling International Conference on Computing Education Research*. 1–10.

[74] Hunter McNichols, Fareya Ikram, and Andrew Lan. 2026. The StudyChat Dataset: Analyzing Student Dialogues With ChatGPT in an Artificial Intelligence Course. In *Proceedings of the LAK26: 16th International Learning Analytics and Knowledge Conference*. 53–63.

[75] Krista L Nelson, Danielle N Newman, Janelle R McDaniel, and Walter C Buboltz. 2013. Gender differences in fear of failure amongst engineering students. *International Journal of Humanities and Social Science* 3, 16 (2013), 10–16.

[76] Sharon Nelson-Le Gall. 1981. Help-seeking: An understudied problem-solving skill in children. *Developmental review* 1, 3 (1981), 224–246.

[77] Richard S Newman. 2012. Adaptive help seeking: A role of social interaction in self-regulated learning. In *Strategic help seeking*. Routledge, 13–37.

[78] Lijun Ni and Mark Guzdial. 2011. Prepare and support computer science (cs) teachers: Understanding cs teachers professional identity. In *American Educational Research Association (AERA) Annual Meeting*.

[79] David G Novick and Karen Ward. 2006. Why don't people read the manual?. In *Proceedings of the 24th annual ACM international conference on Design of communication*. 11–18.

[80] Maria Ong, Carol Wright, Lorelle Espinosa, and Gary Orfield. 2011. Inside the double bind: A synthesis of empirical research on undergraduate and graduate women of color in science, technology, engineering, and mathematics. *Harvard educational review* 81, 2 (2011), 172–209.

[81] Aadarsh Padiyath, Xinying Hou, Amy Pang, Diego Viramontes Vargas, Xingjian Gu, Tamara Nelson-Fromm, Zihan Wu, Mark Guzdial, and Barbara Ericson. 2024. Insights from social shaping theory: The appropriation of large language models in an undergraduate programming course. In *Proceedings of the 2024 ACM Conference on International Computing Education Research-Volume 1*. 114–130.

[82] Miranda C Parker. 2023. Barriers and supports to offering computer science in high schools: A case study of structures and agents. *ACM Transactions on Computing Education* 23, 2 (2023), 1–27.

[83] Lucy Pei and Bonnie Nardi. 2019. We did it right, but it was still wrong: Toward assets-based design. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–11.

[84] Lara Perez-Felkner, Kristen Erichsen, Yang Li, Jinjushang Chen, Shouping Hu, Ladanya Ramirez Surmeier, and Chelsea Shore. 2025. Computing education interventions to increase gender equity from 2000 to 2020: A systematic literature review. *Review of Educational Research* 95, 3 (2025), 536–580.

[85] Prajish Prasad, Rishabh Balse, and Dhwani Balchandani. 2025. Exploring Multimodal Generative AI for Education through Co-design Workshops with Students. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–17.

[86] Thomas W Price, Zhongxiu Liu, Veronica Cateté, and Tiffany Barnes. 2017. Factors influencing students' help-seeking behavior while programming with human and computer tutors. In *Proceedings of the 2017 ACM Conference on international computing education research*. 127–135.

[87] Kevin Pu, Daniel Lazaro, Ian Arawjo, Haijun Xia, Ziang Xiao, Tovi Grossman, and Yan Chen. 2025. Assistance or disruption? exploring and evaluating the design and trade-offs of proactive ai programming support. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–21.

[88] Xiang Qi and Junnan Yu. 2025. Participatory Design in Human-Computer Interaction: Cases, Characteristics, and Lessons. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–26.

[89] Cassie F Quigley, Danielle Herro, Holly Plank, Aileen Owens, and Oluwadara Abimbade. 2024. Transforming computer science education: Exploration of computer science interest and identity of historically underrepresented youth. *Computer Science Education* 34, 4 (2024), 778–805.

[90] Masoumeh Rahimi, Lauren E Margulieux, Dwayne Towell, Jonathan Calver, Dastyni Loksa, and James Prather. 2025. The Impact of Students' Views of Failure on Performance in Introductory Programming Courses. In *Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1*. 611–617.

[91] Zihe Ran, Xiyu Li, Qing Xiao, Xianzhe Fan, Franklin Mingzhe Li, Yanyun Wang, and Zhicong Lu. 2025. How Users Who are Blind or Low Vision Play Mobile Games: Perceptions, Challenges, and Strategies. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–18.

[92] Yolanda A Rankin and Jakita O Thomas. 2020. The intersectional experiences of Black women in computing. In *Proceedings of the 51st ACM technical symposium on computer science education*. 199–205.

[93] Yolanda A Rankin, Jakita O Thomas, and Sheena Erete. 2021. Black women speak: Examining power, privilege, and identity in CS education. *ACM Transactions on Computing Education (TOCE)* 21, 4 (2021), 1–31.

[94] Prerna Ravi, John Masla, Gisella Kakoti, Grace C Lin, Emma Anderson, Matt Taylor, Anastasia K Ostrowski, Cynthia Breazeal, Eric Klopfer, and Hal Abelson. 2025. Co-designing Large Language Model Tools for Project-Based Learning with K12 Educators. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–25.

[95] Yanyan Ren, Shriram Krishnamurthi, and Kathi Fisler. 2019. What help do students seek in TA office hours?. In *Proceedings of the 2019 ACM Conference on International Computing Education Research*. 41–49.

[96] Cecilia L Ridgeway and Tamar Kricheli-Katz. 2013. Intersecting cultural beliefs in social relations: Gender, race, and class binds and freedoms. *Gender & Society* 27, 3 (2013), 294–318.

[97] Monique Ross, Zahra Hazari, Gerhard Sonnert, and Philip Sadler. 2020. The intersection of being black and being a woman: Examining the effect of social computing relationships on computer science career choice. *ACM Transactions on Computing Education (TOCE)* 20, 2 (2020), 1–15.

[98] Mary Beth Rosson, John M Carroll, and Hansa Sinha. 2011. Orientation of undergraduates toward careers in the computer and information sciences: Gender, self-efficacy and social support. *ACM Transactions on Computing Education (TOCE)* 11, 3 (2011), 1–23.

[99] Allison M Ryan and Sungok Serena Shim. 2012. Changes in help seeking from peers during early adolescence: Associations with changes in achievement and perceptions of teachers. *Journal of educational psychology* 104, 4 (2012), 1122.

[100] Milagros Sáinz, Sergi Fàbregues, María José Romano, and Beatriz-Soledad López. 2022. Interventions to increase young people's interest in STEM. A scoping review. *Frontiers in psychology* 13 (2022), 954996.

[101] Elizabeth B-N Sanders and Pieter Jan Stappers. 2008. Co-creation and the new landscapes of design. *Co-design* 4, 1 (2008), 5–18.

[102] Juan Pablo Sarmiento and Alyssa Friend Wise. 2022. Participatory and co-design of learning analytics: An initial review of the literature. In *LAK22: 12th international learning analytics and knowledge conference*. 535–541.

[103] Linda J Sax, Kaitlin NS Newhouse, Joanna Goode, Tomoko M Nakajima, Max Skorodinsky, and Michelle Sendowski. 2022. Can computing be diversified on "principles" alone? Exploring the role of AP Computer Science courses in students' major and career intentions. *ACM Transactions on Computing Education (TOCE)* 22, 2 (2022), 1–26.

[104] Linda J Sax, Kaitlin NS Newhouse, Joanna Goode, Max Skorodinsky, Tomoko M Nakajima, and Michelle Sendowski. 2020. Does ap cs principles broaden participation in computing? an analysis of apcsa and apcsp participants. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. 542–548.

[105] Jeff Schiel. 2024. How High School Students Use and Perceive Technology at Home and School. ACT Research. *ACT Education Corp.* (2024).

[106] Ari Schlesinger, W Keith Edwards, and Rebecca E Grinter. 2017. Intersectional HCI: Engaging identity through gender, race, and class. In *Proceedings of the 2017 CHI conference on human factors in computing systems*. 5412–5427.

[107] Sue Sentance and Andrew Csizmadia. 2017. Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and information technologies* 22, 2 (2017), 469–495.

[108] Narek Shamamyan, Man Su, and Tomohiro Nagashima. 2025. Co-Designing Trustworthy Peer Agents with Middle-School Students. In *Proceedings of the 24th Interaction Design and Children*. 535–544.

[109] Esther Shein. 2019. The CS teacher shortage. *Commun. ACM* 62, 10 (2019), 17–18.

[110] Valerie J Shute. 2008. Focus on formative feedback. *Review of educational research* 78, 1 (2008), 153–189.

[111] James Skripchuk, John Bacher, and Thomas Price. 2024. An Investigation of the Drivers of Novice Programmers' Intentions to Use Web Search and GenAI. In *Proceedings of the 2024 ACM Conference on International Computing Education Research-Volume 1*. 487–501.

[112] Jaemarie Solyst, Shixian Xie, Ellia Yang, Angela EB Stewart, Motahhare Eslami, Jessica Hammer, and Amy Ogan. 2023. "I would like to design": Black girls analyzing and ideating fair and accountable AI. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–14.

[113] John Stamper, Ruiwei Xiao, and Xinying Hou. 2024. Enhancing llm-based feedback: Insights from intelligent tutoring systems and the learning sciences. In *International Conference on Artificial Intelligence in Education*. Springer, 32–43.

[114] John Sweller. 2020. Cognitive load theory and educational technology. *Educational technology research and development* 68, 1 (2020), 1–16.

[115] Xiaohang Tang, Sam Wong, Marcus Huynh, Zicheng He, Yalong Yang, and Yan Chen. 2025. SPHERE: Supporting Personalized Feedback at Scale in Programming Classrooms with Structured Review of Generative AI Outputs. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. 1–17.

[116] Tiera Tanksley, Angela DR Smith, Saloni Sharma, and Earl W Huff Jr. 2025. "Ethics is not neutral": Understanding Ethical and Responsible AI Design from the Lenses of Black Youth. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–20.

[117] Karen W Tao and Alberta M Gloria. 2019. Should I stay or should I go? The role of impostorism in STEM persistence. *Psychology of Women Quarterly* 43, 2

(2019), 151–164.

[118] Jakita O Thomas, Nicole Joseph, Arian Williams, Chan'tel Crum, and Jamika Burge. 2018. Speaking truth to power: Exploring the intersectional experiences of Black women in computing. In *2018 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. IEEE, 1–8.

[119] George N Triantafyllakos, George E Palaigeorgiou, and Ioannis A Tsoukalas. 2008. We! Design: A student-centred participatory methodology for the design of educational applications. *British Journal of Educational Technology* 39, 1 (2008), 125–139.

[120] Daphne Van Zandvoort, Marloes Vredenborg, and Marit Bentvelzen. 2025. Unhealthy Comparisons to Promote Healthy Behavior? Exploring the Impact of Social Comparison Strategies in Personal Informatics.. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–21.

[121] Lev S Vygotsky. 1978. *Mind in society: The development of higher psychological processes*. Vol. 86. Harvard university press.

[122] Karla Waldenmeier, Susanne Poeller, Martin Johannes Dechant, Nicola Baumann, and Regan L Mandryk. 2024. Cheat Codes as external support for players navigating fear of failure and self-regulation challenges in digital games. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–13.

[123] Jennifer Wang, Hai Hong, Jason Ravitz, and Sepehr Hejazi Moghadam. 2016. Landscape of K-12 computer science education in the US: Perceptions, access, and barriers. In *Proceedings of the 47th ACM technical symposium on computing science education*. 645–650.

[124] Jayce R Warner, Carol L Fletcher, Nicole D Martin, and Stephanie N Baker. 2021. Applying the CAPE framework to measure equity and inform policy in computer science education. *Policy Futures in Education* (2021), 14782103221074467.

[125] Timothy J Weston, Wendy M Dubow, and Alexis Kaminsky. 2019. Predicting women's persistence in computer science-and technology-related majors from high school to college. *ACM Transactions on Computing Education (TOCE)* 20, 1 (2019), 1–16.

[126] Steve Whittaker, Loren Terveen, and Bonnie A Nardi. 2000. Let's stop pushing the envelope and start addressing it: a reference task agenda for HCI. *Human–Computer Interaction* 15, 2-3 (2000), 75–106.

[127] Destiny Williams-Dobosz, Amos Jeng, Renato FL Azevedo, Nigel Bosch, Christian Ray, and Michelle Perry. 2021. Ask for help: Online help-seeking and help-giving as indicators of cognitive and social presence for students underrepresented in chemistry. *Journal of Chemical Education* 98, 12 (2021), 3693–3703.

[128] Rainer Winkler and Julian Roos. 2019. Bringing AI into the classroom: Designing smart personal assistants as learning tutors. (2019).

[129] Greta Winograd and Jonathan P Rust. 2014. Stigma, awareness of support services, and academic help-seeking among historically underrepresented first-year college students. *Learning Assistance Review (TLAR)* 19, 2 (2014).

[130] Marisol Wong-Villacres, Aakash Gautam, Deborah Tatar, and Betsy DiSalvo. 2021. Reflections on assets-based design: A journey towards a collective of assets-based thinkers. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–32.

[131] Marisol Wong-Villacres, Arkadeep Kumar, Aditya Vishwanath, Naveena Karusala, Betsy DiSalvo, and Neha Kumar. 2018. Designing for intersections. In *Proceedings of the 2018 Designing Interactive Systems Conference*. 45–58.

[132] Ryoko Yamaguchi and Jamika D Burge. 2019. Intersectionality in the narratives of black women in computing through the education and workforce pipeline. *Journal for Multicultural Education* 13, 3 (2019), 215–235.

[133] Stephanie Yang, Hanzhang Zhao, Yudian Xu, Karen Brennan, and Bertrand Schneider. 2024. Debugging with an AI tutor: Investigating novice help-seeking behaviors and perceived learning. In *Proceedings of the 2024 ACM Conference on International Computing Education Research-Volume 1*. 84–94.

[134] Chengbo Zheng, Kangyu Yuan, Bingcan Guo, Reza Hadi Mogavi, Zhenhui Peng, Shuai Ma, and Xiaojuan Ma. 2024. Charting the future of AI in project-based learning: a Co-design exploration with students. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–19.