This is an Accepted Manuscript of an article published by Taylor & Francis in *Computer Science Education* on August 18th, 2020, available online:
https://www.tandfonline.com/doi/full/10.1080/08993408.2020.1805286

Reference: Dahn, M., & DeLiema, D. (2020). Dynamics of emotion, problem solving, and identity: Portraits of three girl coders. *Computer Science Education*.
https://doi.org/10.1080/08993408.2020.1805286

Dynamics of Emotion, Problem Solving, and Identity: Portraits of Three Girl Coders

Maggie Dahn[a]* and David DeLiema[b]

[a] *Education and Informatics, University of California, Irvine, Irvine, USA*

[b] *Educational Psychology, University of Minnesota, Minneapolis, USA*

Both authors contributed equally to this paper

*Corresponding author: Maggie Dahn; maggiedahn@gmail.com/dahnm@uci.edu; 630-650-0588

Biographical notes

Maggie Dahn is a postdoctoral scholar at the University of California, Irvine working with the Connected Learning Lab and Creativity Labs. She received her PhD from the Urban Schooling Division of UCLA's Graduate School of Education and Information Studies in 2019 with support from an NAEd/Spencer Dissertation Fellowship. She engages in design research to study how people learn through conversation and interaction. She is interested in how art making is related to voice and identity development and the role of emotion in learning.

David DeLiema is an Assistant Professor in the Department of Educational Psychology at the University of Minnesota. The long-term goal of his research is to understand, critically examine, and re-imagine how instructors and students notice, justify, and respond to moments of failure during learning. Toward this end, his work covers topics such as playful learning, embodied cognition, spatial reasoning, storytelling, and productive failure, often within technology-rich settings and always foregrounding the process of learning through social interaction.

Dynamics of Emotion, Problem Solving, and Identity: Portraits of Three Girl Coders

Abstract

Background and Context: Women are underrepresented in the field of computer science, a trend that in part can be traced to girls' early experiences with the discipline.

Objective: Our aim is to show how three girls who became strong coders talked about their debugging practice at the intersection of problem solving, emotion, and identity.

Method: We use the portraiture methodology to trace the *goodness* of a designed programming workshop environment. We aim to show the trajectories of three strong coders over the course of two years of participation in weekend and summer workshops.

Findings: We found that creative reflection spaces through journaling, art making, and storytelling opened possibilities for the learners to observe, understand, and critically examine the integration between problem solving, emotion, and identity in their programming experience.

Implications: Findings have implications for designing inclusive programming learning environments that invite collective reflection on the moment-to-moment experience of learning to code.

Keywords: computer science, debugging, failure, portraiture, identity

Dynamics of Emotion, Problem Solving, and Identity: Portraits of Three Girl Coders

Introduction

It is well documented that women are underrepresented in the computer science field (Ashcraft et al., 2012; Lehman et al., 2016; Pantic & Clarke-Midura, 2019). In many cases, this noted underrepresentation can be traced to early experiences with coding that shape learners' identifications in relation to computer science (Friend, 2015). Researchers have suggested that an early "experience gap" between boys and girls in computing may contribute to girls reporting lower confidence in their abilities (Hur et al., 2017; Margolis & Fisher, 2003), making them less likely to see themselves as capable of pursuing computer science careers (Lewis et al., 2016). A contributing factor is that computer science learning environments may not be designed with girls in mind, thus presenting missed opportunities for early engagement with computer science (Lyon & Green, 2020). Attending in detail to girls' early computer science experiences with an eye toward facets of marginalization can inform future designs of learning environments that aim to fully and equitably support girls' explorations of computer science.

Toward this end, we have focused this paper on girls' early experiences with computing by following the trajectories of three strong coders over the course of two years of their participation in weekend and summer computer science workshops that we co-designed with non-profit workshop coordinators and often taught ourselves. The workshop design, which we describe in detail in previous work (Dahn et al., 2020; DeLiema et al., 2020), drew on principles of productive failure (Kapur, 2008), design elements of reciprocal teaching (Palincsar & Brown, 1984), and known debugging strategies (Zeller, 2009), with the aim of fostering a supportive culture around *bugs* or *errors* that are frequent and challenging facets of programming. These workshops supported a debugging culture through other purposeful design choices, including peer mentorships, tools for debugging, and constructivist, project-based approaches to coding curriculum and pedagogy. Here we focus on a novel facet of the design—the ways we asked learners to creatively reflect on their coding and debugging experiences through journaling, art making, and storytelling. The literature confirms that it is important for girls to have frequent and in-depth experiences with programming to better understand and build confidence within the discipline (Dasgupta & Stout, 2014; Hur et al., 2017; Master et al., 2016). Through our study, we argue that fostering reflection on how it feels to engage with the practice of computer programming matters for building this confidence, which in turn, shapes learning and identity.

We were interested in how learners—in particular strong girl coders—made sense of their experiences during moments of struggle with coding. Our guiding research question was: *How do three middle and high school girls who become strong coders reflect on problem solving during moments of struggle with respect to the processes and strategies they use, their emotions, and their identities?* We drew on the portraiture methodology (Lawrence-Lightfoot & Davis, 1997; Lawrence-Lightfoot, 2016) to document their trajectories and argue that creative reflection spaces opened possibilities for learners to observe, understand, and critically examine the integration between their experiences with problem solving, emotion, and identity as they were learning to program.

Centering debugging

Why focus on debugging to study learners' early experiences with problem solving, emotion, and identity? In short, moments of failure in programming provide likely settings in which these three dimensions intersect. Because debugging code involves locating what went wrong, generating conjectures about what caused it, and experimenting with possible fixes, all while choosing from an array of tools and strategies, debugging requires *problem solving* (McCauley et al., 2008). In addition, in line with general predictions that "contradictions, incongruities, anomalies, obstacles to goals, and other impasses" (D'Mello & Graesser, 2012, p. 145) are sites for uncertainty and reflection (Koschmann et al., 1998), debugging is now increasingly recognized as generating layered *emotional experiences* (Dahn et al., 2020; Kinnunen & Simon, 2010), especially feelings of confusion, frustration, and boredom among newcomers (Bosch et. al, 2013). In this same way, in line with observations that moments of failure invite statements about who is at fault (Weiner, 1985), statements about one's efficacy to navigate a problem (Bandura, 1982), and possible inequities in interactions with peers (Shah & Lewis, 2019), experiences with debugging may profoundly shape the social space in which learners' *identities* form in relation to programming.

Emotion in the context of learning to program

In articulating a vision for a "more inclusive computer culture" more than 25 years ago, Turkle and Papert (1992) reframed how programmers interact with computers through a "set of intellectual *and* emotional values" (p. 20, our emphasis). Simply put, learning to code can be an intense and even visceral emotional experience (Dahn et al., 2020; Kinnunen & Simon, 2010). Furthermore, emotion is not just an embodied sensation, but also part of the process of deciding what to blame for failures (Weiner, 1985) and how to plan a response (Oatley, 1987). Emotion is embedded in the problem solving process (DeBellis & Goldin, 2006) and serves as a "driver" of decision making (Lerner et al., 2015, p. 799). Relevant to our argument, emotions impact how learners understand themselves and their identities (Heyd-Metzuyanim, 2013), described by Lemke (2010) as a dynamic process of "identification and dis-identification."

For these reasons, it is not surprising that researchers have started to recognize socioemotional facets of learning to program (Bosch et al., 2013; Bennedsen & Caspersen, 2008), including how emotions occur at different stages of programming assignments (Kinnunen & Simon, 2010). Emotion may also matter for the quality of programming collaboration. In a high school pair programming context within a physical computing curriculum, the quality of social interaction depended in part on learners' impression that their partners were emotionally supportive (Lui et al., 2020). Related computer science education research has adopted a phenomenological lens on how it feels for newcomers to learn to code, attending to "discursive, perspectival, material and embodied experiences" (Sengupta et al., 2018, p. 49; see also Kinnunen & Simon, 2012). We extend these analytical threads with research that examines emotion among middle and high school students. Because gender-based assumptions about students' STEM capacities are well-documented in middle and high school (Ramirez et al., 2011), this developmental window is a critical time to investigate the complexities of problem solving, emotion, and identity.

Girls' identities within computer science

Researchers have attended to how learners construct identities within computing (e.g., Rodriguez & Lehman 2017; Tonso, 2006; Wong, 2016). We consider identity within computing to be a social phenomenon influenced by dominant identity discourse and how young people see themselves fitting within current social constructions (Wong, 2016). Identification with computing is also impacted by the organization of the immediate learning environment, including how programming activities are designed to support social recognition by peers and teachers so that young people are positioned as experts (Fields

& Enyedy, 2013). As such, we take up the idea that "identity is not a static 'thing' but rather a dynamic and changing process over time," (Fields & Enyedy, 2013, p. 17), part of a *trajectory of identification* (Wortham, 2006), malleable and responsive to the design of learning spaces (Wong, 2006).

Because we are interested in domain-specific identities related to coding and understand identities as taking shape in the figured world of the classroom (Holland et al., 1998), we draw from Nasir and Hand (2008) who conceive of identities as practice-linked, or identities that live within the connection between self and activity bound to the social and cultural practices with which learners engage. This includes how accessible the domain is to a participant, the roles a participant takes on, and the ability to feel confident and make contributions. As part of a practice-linked identity, we see identities as endorsable stories learners tell about themselves that imply membership within communities (Sfard & Prusak, 2005). Our conception of computer science identity also includes the element of time in which learners acknowledge past, present, and future selves (Brickhouse & Potter, 2001). Like Hand and Gresalfi (2015), we view the identities youth develop as ways they make "social futures" (p. 201) for themselves and think about their potential for long-term identification with computer science.

Research suggests that the more experiences women have shaping the learning environment and contributing to part of a computer science culture, the more they feel belonging (Frieze et al., 2012). Studies of college-aged learners have attempted to increase retention of women in computer science majors through targeted interventions (e.g., Dekhane & Napier, 2017). Lehman et al. (2016) suggest that addressing the gender gap can be supported by designing for the unique characteristics of women successfully pursuing computer science careers, which included higher self-ratings of artistic ability when compared to men. For young girls, e-textiles have shown great promise for challenging traditional gender roles in computing (Buchholz et al., 2014; Peppler, 2016). Cultivating a sense of belonging can be supported by pedagogy that centers lived experiences of youth (Pinkard et al., 2017). Similarly, Tissenbaum et al. (2019) suggest that computer science learning environments ought to be designed around the notion of *computational action*, meaning that youth should have opportunities to engage in computing activities that directly impact their lives and communities.

We focused on the immediate coding and debugging experiences of learners to understand their practice-linked identities. We looked at strong girl coders to document how our design of creative reflection spaces—including art making, storytelling, and journaling—worked for them. We were interested in understanding more about how these learners saw themselves, why they were successful, and how our designs supported their positive development.

Our focus learners identify as Latina girls. While white girls pursue science disciplines at nearly the same rate as boys, those from minoritized ethnic groups have not (National Academies, 2007; National Center for Education Statistics, 2011). Rodriguez and Lehman (2017) write about the need to theorize computing identity as an intersectional concept to understand how layers of race, ethnicity, and gender impact how learners understand themselves within computer science. Unpacking the possibilities for identity in computer science is important for adolescent girls, especially girls of color, who are often discouraged to engage or persist with computer science due to stereotypes and socialized beliefs about computer scientists (Varma, 2010). While the focus girls' Latina identity was not explicitly incorporated within our curriculum design, nor did it emerge explicitly in how they talked about their problem solving processes, it is important within the context of broader narratives about who gets to engage in computer science. Noting the limitations of our design, we acknowledge this study is situated within what some scholars have called *identity work* (Calabrese Barton et al., 2013), in which learners position themselves in relation to historical and cultural understandings of practice.

Dynamics of emotion, problem solving, and identity in computer science

To focus our theoretical lens (see Figure 1), we draw from scholars who discuss the dynamic relationship between emotion, problem solving, and identity in domain-specific ways. One example comes from Heyd-Metzuyanim's (2015) case study of a middle school girl's relationship with mathematics. Heyd-Metzuyanim presents a framework for studying the close connection between learning, emotion, and identity, describing how the three interact with cognitive skills required to engage with mathematics. In her trace of an individual learning trajectory, she notes the interactions between complex layers of emotion, cognition, and how a learner's identity develops, arguing that they are constitutive of one another and inseparable.

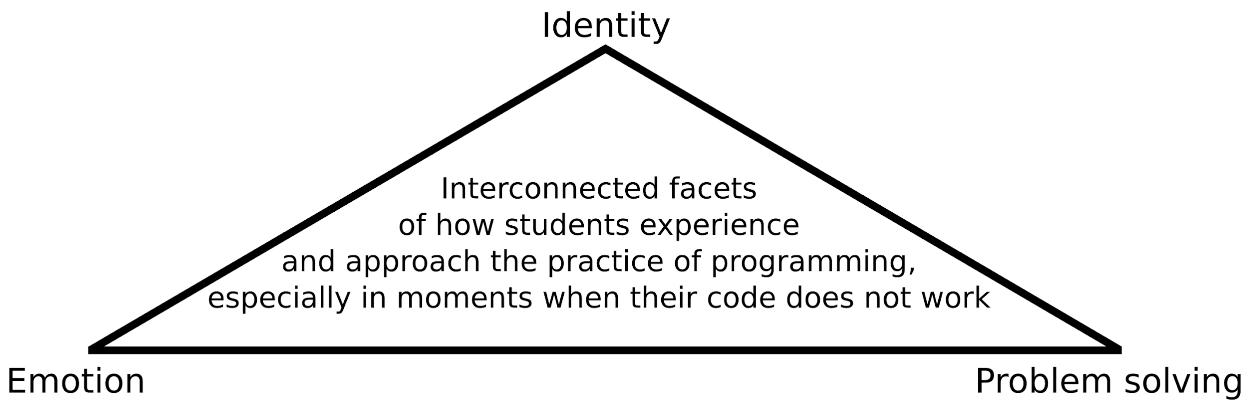

Figure 1. Interconnected layers of emotion, problem solving, and identity in programming

By looking at the intersection of emotion, problem solving, and identity, we can get a sense of who learners are at particular points in time as they learn to code, which might reveal something deeper about how they see themselves in relation to computer science. Aligned with work describing inequities in computer science education (e.g., Margolis et al., 2012; Nasir & Vakil, 2017), we argue that a stronger understanding of what learners feel and think, as well as how they see themselves, will better equip researchers and teachers to design more inclusive learning spaces.

Portraiture to trace the *goodness* of an environment

We drew from the portraiture methodology to construct narratives about three focus learners because we were aiming to understand success, or what a portraitist would call the *goodness* (Lawrence-Lightfoot, 1986) of a learning environment. Kinnunen and Simon (2012) argue that qualitative methods including phenomenography and grounded theory are useful tools for computer science education researchers, and we think that portraiture adds another qualitative lens that attends closely to learner experience.

Lawrence-Lightfoot and Davis (1997) explain that as the portraitist mines for *goodness*, she focuses on the positive in relation to a phenomenon of interest. Because we noticed that our focus learners were expressing facets of their identities and positively identifying as problem solvers within their coding practice over time, we determined that portraiture would support us in uncovering the goodness of practice and how the designed environment—including the creative forms of reflection we made central––provided a backdrop for their positive identification.

The portraiture methodology includes five central features: context, voice, relationship, emergent themes, and aesthetic whole. To elaborate, the context in which participants are embedded shapes all experience; the voice of the researcher is evident in the interpretive work and voices of participants are foundational data sources for constructing portraits; the relationship between the portraitist and participants supports the dialogic construction of portraits; emergent themes lend analytical coherence to the interpretive work of portraiture; and finally, how portraits are presented as an aesthetic whole stress an artistic process.

The portraitist has a central voice as "narrative text is blended with the interpretive perspective of the researcher combining to create a shared world of understanding" (Cope et al., 2014, p. 89). In our case, we were instructors and researchers involved in the details of curriculum design, teaching, and data collection. We aim to be explicit that our findings are a result of our collaborative, interpretive work, presented as they are not in spite of, but precisely because of the relationships we had with participants and data.

Method

Overview of setting and intervention

We used design-based research (Brown, 1992; Collins, 1992) to create workshops at a computer science education non-profit for 5th through 10th graders. The non-profit ran a variety of computer science education classes at schools and through summer, weekend, and after school programming. Our workshop focused on surfacing learners' reflections on failure. Curriculum consisted of a progression of coding concepts with interleaved skills-based and project-based modules emphasizing debugging (see DeLiema et al., 2020 for details). Participants wrote code in JavaScript and Lua (on software platforms including Minecraft, PixelBots, OpenProcessing, and Lego Mindstorms) to create beat machines, mosaic paintings, robot dances and battles, video games, and avatars to use in a live Dungeons and Dragons game.

We use the portraiture methodology to report on the individual experiences of three strong girl coders who participated consistently over 26 months of weekend and summer coding workshops between summer 2016 and summer 2018 (see Table 1). The iterative component of our design-based research resulted in some specific design changes from the summer 2017 workshop to the one in summer 2018. For example, in summer 2017, learners rotated through three coding classes with different instructors and participated in a related art class over two weeks (see Dahn et al., 2020, for details on art making projects). In 2018, the summer workshop ran for three weeks to support deeper participation, and learners remained in one classroom with the same 2-3 instructors who taught both coding and art. During both workshops, we conducted semi-structured interviews with all participants about their coding, debugging, and art making experiences (see protocols in appendix of Dahn et al., 2020).

The second author had been part of the non-profit community since 2013 as an instructor and researcher. In 2017 the first author was the art teacher for all 63 participants in the summer workshop, and in 2018 both authors co-taught coding and art with an additional coding instructor in a single classroom of 21 participants. The first author was formerly an elementary school art teacher with no coding experience, and the second author had worked as a coding instructor at the affiliated non-profit.

Focus learner selection

Focus participants, all three of whom attended Title I schools and identified as Latina, were chosen based on the depth of their coding practice, their commitment to their coding and artwork, their attendance at both summer workshops, and availability of data. By the final summer workshop these three participants had learned fundamental programming concepts (e.g., sequences, loops, conditional statements, functions, multi-parameter functions, and arrays), were consistently applying these concepts in self-directed and independent ways, and were reliable mentors.

Table 1. *Girls' attendance at weekend and summer workshops at non-profit*

	Win 2017	**Spr 2017**	**Sum 2017 ***	**Fall 2017**	**Win 2018**	**Spr 2018**	**Sum 2018 ***
Lorena	X	X	X	X	X	X	X
Alexys	X	X	X	X	X	X	X
Kat	X	X	X	X	X	X	X

*notes summer workshops, during which the majority of data were collected; summer 2017 ran for 2 weeks (6 hours/day) and summer 2018 ran for 3 weeks (6 hours/day)

Data sources

Relevant data include interview transcripts, artwork, written artist statements, journal entries, and video transcripts of in-class storytelling with peers. Interviews were conducted by authors. Because the interviews most straightforwardly conveyed how learners experienced debugging, we relied on them most when first crafting our portraits, yet we triangulated findings with other sources (Sandoval, 2012). We incorporated data from journals, artist statements, and in-class storytelling to add texture to the stories girls mapped out in interviews. We approached this work with awareness that a unified picture was not guaranteed (Tracy, 2010). We knew that a coherent/convergent picture might emerge across *triangulated* data sources (Denzin, 1978), or that a fragmented, divergent picture might emerge across data sources, what in postmodern frameworks has been described as *crystallization* (Richardson, 1997). We discussed the possibility of including video records of what participants said and did during coding, but we lacked resources for such a large initiative. In addition, in reflecting on the rigor of our approach, we agreed that the focus on participants' reflections on their coding practice, our longitudinal data set spanning two years, and our heterogeneous data sources provided enough substance to detail how learners viewed their own problem solving processes, emotions, and identities.

Analytical approach

We first re-reviewed and transcribed all data. Because we were interested in how learners talked and wrote about their coding strategies and processes, emotions, and identities, our initial codes considered these elements in isolation. However, we came to understand that emotion was intertwined with learners' talk about their problem solving processes and identities. More directly, we found that providing a space for reflection on the emotional experience of learning to code supported learners in seeing themselves as the kinds of people who could solve problems.

Our operationalization of identity was aligned with Heyd-Metzuyanim's (2015) conceptualization of identity as explicit talk about a person. To glean statements about identity from learners' talk, in our data we pay attention to specific statements, generalized statements about recurring experiences, and stable traits (Heyd-Metzuyanim, 2015). These statements include remarks about how learners participate and how they evaluate their own membership within a particular group. To operationalize emotion, we drew on our prior work (Dahn et al., 2020) in which we used a storytelling framework (Herman, 2009) to attend to descriptions of internal experience about debugging. We understand emotion in the body as “multifaceted, biologically mediated…(experiential, cognitive, behavioral, expressive)” (Lerner et al., 2015, p. 800), but also a vibrant part of the social world in which we communicate *about* emotion (Goodwin et al., 2012). In this way, we attended to a range of dimensions of emotions, including the novelty of description, change over time, and emotions about emotions. Comparable to work in discursive psychology that examines internal experience as part of communication in a social context (Wiggins, 2016), we viewed girls’ reflections on emotion as dynamically coupled to other expressions they made about their coding process. The final construct we considered, problem solving, was operationalized according to how participants described the tools (e.g., syntax checkers, steppers) and strategies (e.g., isolate the problem, experiment with a fix) they used to debug problems in their code. We included practices from computer-science debugging literature (e.g., McCauley et al., 2008; Zeller, 2009) and additional practices participants named as valuable.

We wrote short narratives and pulled out resonant and repetitive refrains from girls’ interviews. From these we inductively developed themes to capture how reflection on emotions supported the ways learners talked about themselves as problem solvers. We aimed to tell a fluid story of each girl while still maintaining the systematicity and precision we began with in looking at the particulars of problem solving, emotion, and identity.

Lawrence-Lightfoot (1986) argues that the portraitist should work as an ethnographer, fully embedding herself in the research setting. As we note above, we spent time with focus learners as their instructors, as researchers, as co-creators of artwork, and as people chatting over lunch. Given these multifaceted roles and participation frameworks, we aimed to be self-reflexive in analysis, challenging one another’s inferences and transparently documenting our rationales (Tracy, 2010). In addition to drawing from the portraiture methodology to construct narratives, we used the constant comparative method (Glaser, 1965), developing themes through explicit definitions of constructs, iterative examinations of data, and memos documenting evidence for claims. Using a constant comparative method paired with portraiture, we coded data around themes of problem solving, identity, and emotion, comparing new instances of data to prior codes to refine each theme.

Findings

Lorena

Lorena was a rising 10th grader during the summer of 2018. As her instructors, our impression was that in both art and coding, she took her time, paid close attention to details, and held herself to a high standard. She would sometimes take art projects home to finish them. Lorena also spent some breaks and lunch periods chatting extensively with instructors about new coding concepts. In her artwork, she documented having conversations about code with peers that continued into break and lunch periods (Figure 2). In all, Lorena took responsibility for her own learning, and in both art and coding, seemed to take pride in her work.

In building a portrait of Lorena, it became clear that she had mixed reactions about her approach to coding. The following journal entry after a difficult day of coding in 2017 hinted at this dynamic: “[I] have to work at home on code now :) ‘great’—sarcasm.” Her commitment to working at home, with a smile, was hedged with a sarcastic “great.” In a similar vein, one of her works of art (Figure 2) used plants drawn with fine details as an anchoring metaphor for growth. However, the vines showing growth also tracked the “# of runs until code was correct.” This recognition of *extensive personal effort* to reach correct code, especially through multiple attempts at formulating a solution, balanced with an *awareness of growth*, is what we unpack in the portrait below.

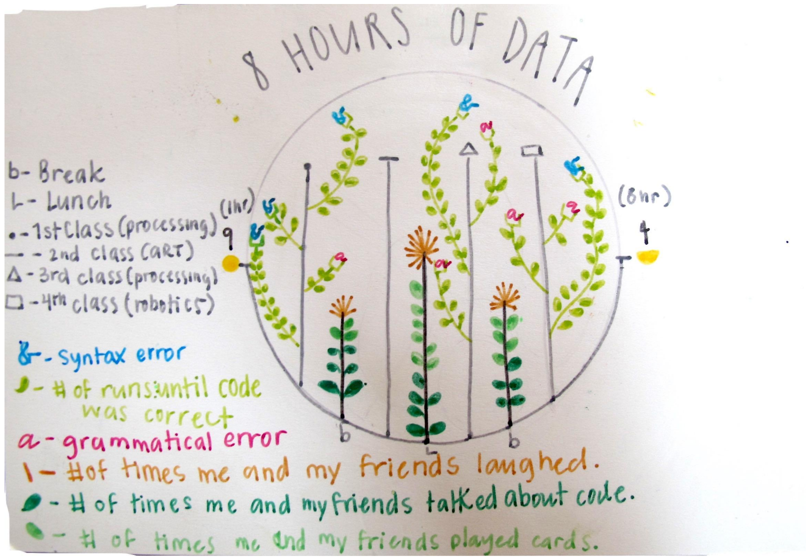

Figure 2. Lorena’s data visualization artwork

Overthinking

When describing her debugging process, Lorena frequently alluded to “overthinking” solutions and moving slowly. Lorena explained how she liked to have a sense of control over what she did, was most content when things were organized, and emphasized the value of being aware of her feelings as “a good thing so you could change them if they’re negative.” Lorena acknowledged that her desire to understand problems led her to persistently read her code:

> “didn’t reread my code
> #reread my code
> #reread my code more” (Journal Entry, Winter 2017)

Lorena further explained in an interview that she sometimes “overcomplicates” things by taking up “way too much code line” when writing code. She also linked her experience of overthinking to a general statement about her speed: “I guess I’m kind of slow, which kind of sucks because I overthink it, the solutions.” And in a journal entry on “what to know when coding” (Figure 3), Lorena advised herself to “think about the problem and be aware of the issue.”

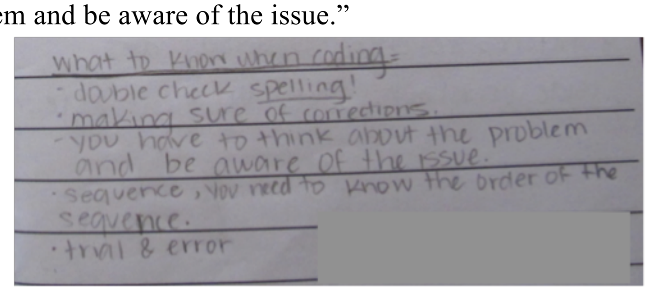

Figure 3. Lorena’s journal entry on “what to know when coding”

Being someone who “overthinks,” is “aware of the issue,” “#rereads my code more,” and is “slow” and controlled, which Lorena viewed as something that “kind of sucks,” is how she often saw herself in relation to her problem solving practice.

Overthinking in comparison to others

Lorena consistently compared her observations about overthinking and self reliance to her peers. Using her abstract watercolor artwork from Summer 2018 as a springboard for reflection (Figure 4), she noted a stark comparison between herself and others:

> This is me debugging [referencing the bottom of Figure 4] and this is how I feel like everyone else debugging [referencing the top of Figure 4]. We both encounter a bug and it's the same bug, but with me since Mia says I'm meticulous and overthink a lot of things, I overthink the solution and I don't follow it logically like other people, which we both eventually like reach the solution but I have a hard time navigating my way, unlike other people.

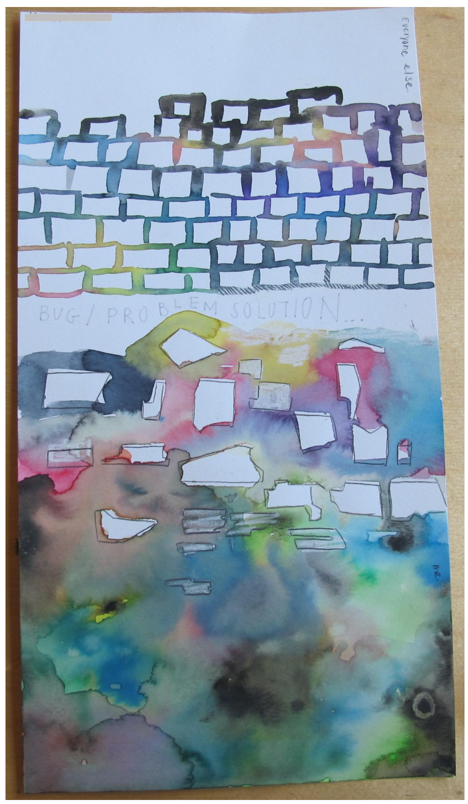

Figure 4. Lorena’s abstract artwork

Drawing a contrast with her peers, Lorena positioned herself as someone who overthinks, does not follow a logical path, and has difficulty navigating her way to a solution. She elaborated that she tried to “find different solutions,” and that her code was sometimes “overcomplicated,” voicing the words of her instructor, Mia, when reifying this perception of her problem solving identity. In her artist statement for Figure 4, Lorena described this gap in a positive and negative light:

> When finding a bug in code I noticed how fast or "easy" other coders solved the problem...I on the other hand seemed to over complicate/overthought solutions leading me to find the alternate solution. Although overthinking is a bad habit it helps me see the same situation in a different perspective. The right side represents me...unorganized thoughts...scattered...still eventually reaching the "solution". Whereas the right side represents fellow coders and everyone else in life...organized simplistic...getting to the solution more logically.

In her interview, Lorena said that this "bad habit" seemed to "put [herself] down," and yet her artist statement hinted at a strength, that overthinking "helps [her] see the same solution in a different perspective...still eventually reaching the 'solution'." This double bind was further reflected in Lorena consistently nudging herself to ask for help. In a journal entry from winter 2017, she implored herself to "#ASK PLEASE," and then the next day to "#ASK!" A few months later she wrote to remind herself to "ask for help and explanation if I don't understand something," "attempt to explain more clearly when asking for help," and to "not freak out, ask for help." Lorena noted that "working independently didn't really help. But did." Those final words ("but did") capture Lorena's mixed perception of her self-reliance and overthinking as something that worked but also "kind of sucked."

Past and present versions of self

When Lorena reflected on her past and present experiences of debugging, the themes of overthinking and comparing herself to others came together and suggested a shift in emotion and strategy over time. She explained:

> I used to get really embarrassed if I couldn't pass a bug, and I would just look over at other people's screens and I would be like oh jeez, like i'm behind. Now I'm more calm about it like I just try to follow the strategies, and if I can't get that specific bug I'll just move on to something else or I'll comment whatever is making it wrong.

Lorena described trying to "pass a bug" as something that "used to get [her] really embarrassed," prompting her to look at "other people's screens," and feel, "oh jeez, like I'm behind." In a 2017 journal entry, she echoed this perspective by urging herself to "stay calm" and "solve one problem first (bug) instead of moving on to the next." In contrast, Lorena described her present approach to coding: "more calm about it," "follow[ing] the strategies," and "if I can't get that specific bug I'll just move on to something else." These reflections represent a change over time in how she approached bugs; while she would first solve bugs one at a time, get embarrassed, and feel behind, she eventually found ways to keep herself moving as she solved bugs, including following debugging strategies, staying calm, and returning to problems after a break. In looking toward the future when solving bugs in code, Lorena explained that perhaps she should "not stress [herself] out too much and think of easier solutions."

In summary, Lorena's reflections on emotion, identity, and problem solving were inextricably linked. With the phrase, "I'm kind of slow, which kind of sucks because I overthink it, the solutions," Lorena bridged a way she saw herself ("I'm kind of slow") with an emotional valence ("kind of sucks") and an approach to finding solutions ("I overthink it"). In her artist statement and interview, Lorena described her identity as someone who overthought and was slow, yet even in naming these qualities she also socially positioned herself in a positive light as someone who approached problems differently and saw things from a different perspective. Lorena's story had an arc to it. She started by describing herself as self-reliant, scattered, behind, and having a hard time, accompanied by an emotional resonance of freaking out, embarrassment, and feeling like it sucks. Lorena described her initial problem solving practices, including re-reading code, thinking about the problem, sticking with the bug until it was finished, and wanting to ask for more help, as practices that "did" work because she could still reach the "solution." Over time, Lorena described an overarching shift toward feelings of calmness when faced with bugs and a problem solving approach in which she utilized debugging strategies and temporarily moved on from troublespots. Lorena's reflections point to the connected nature of identity and emotion that mutually informed one another during her problem solving process. As Lorena came to excel as a coder over this two year period, repeated opportunities for reflection that were embedded in our workshop design (e.g., journal entries over time, written artist statements) nudged her simultaneously to articulate out loud and in detail who she was as a coder and how she was changing in ways significant to her.

Alexys

Alexys was a rising 10th grader during the summer of 2018. Her younger brother, who was in sixth-grade, was also part of the workshop. Because he was less experienced, her brother sometimes asked her for help with his code, and she readily offered assistance. In reflecting through artwork, Alexys created pieces that she described as related to both her coding experience and more general themes related to living in the world. For example, she explained in an artist statement that her abstract art in Figure 5 shows the positive and negative sides of a person's day in coding and in life.

Figure 5. Alexys's abstract artwork

In our portrait of Alexys, we highlight how she described a shift in confidence as she gained more debugging and coding experience, in particular by growing her repertoire of debugging strategies. In turn, she began to identify as a mentor to others. She also emphasized being a creative person in coding and art.

Shifting confidence linked to growth in problem solving strategies

Like Lorena, Alexys talked about her problem solving process in terms of how it changed over time. Alexys drew a contrast between her prior reactions to bugs and how she handled them as a more experienced coder. In her interview from the first summer workshop, she explained that “before like last summer or like weekends I didn't really know how to debug until like now,” describing an arc of progress from “you don't know what to do so you kind of like freak out and say ‘where's the bug at?’” toward “knowing” how to debug. She unpacked “knowing” across interviews by describing a range of debugging strategies in her toolkit: “debugging you have to go step by step,” check for “spelling,” check “how you put it together,” use the “console at the bottom, it usually tells you what is wrong,” “reread my whole code,” “see if anything's missing,” “double check,” “revise what is wrong,” and “go through my past work.”

Overall, Alexys described learning to better understand and deal with what she was “stressing about.” In her interview from the second summer workshop she explained, “My first like two years I’ve been here I would get like a lot of bugs and stuff and I'd always ask for help,” adding that “if you're like stressing over something and you draw it out you could see like what you're stressing about and like maybe you could ask someone like how you could fix it.” Here Alexys linked a feeling of stress, what she characterized in the first summer workshop interview as an “explosion,” with dependence on others to debug her code. Later in the interview, she explained how her feelings shifted over time: “Since I pretty much know how to debug already I don't really like get nervous about it so I just kind of go with it.” She attributed this change from nervousness to “going with it” to learning more about how to debug and to her growing confidence as a coder and independent problem solver. By the second summer, Alexys viewed debugging and coding as connected in such a way that you could not have one without the other: “I think debugging....if you didn’t have it, I don’t know what like the whole point of coding [would be],” and then emphasized how learning necessarily derives from mistakes. This shift in confidence supported Alexys as she took on a mentorship role and began to see herself as someone who could help others, a theme we elaborate below.

Becoming a mentor

Importantly, gaining confidence in her abilities as a coder helped Alexys to see herself as a mentor for her younger brother and other novice coders. She explained, “It just feels good that they know that you know all the stuff and yeah it just feels great like helping out.” Through this statement she positioned herself as a mentor, as someone who others could turn to for help because she “know[s] all the stuff.” She acknowledged that this recognition as a mentor made her feel “great like helping out.” Alexys then explained her motivation for wanting to be a mentor to younger, novice coders as well as the kind of mentor she hoped to be:

> I actually wanted them to do some codes because I’ve actually been here four years so wanted to see how the little kids were actually using their minds to create code so if they ever need help I would actually help but I just wanted them to be creative because I would probably do something more like experienced but they’re here new so I was just like no you do it... and act all bossy to them no... and also I’m not that controlling, I just don’t like being the leader, that’s just not me.

Alexys described wanting to understand how novice coders (whom she called “little kids”) were creating code on their own before she determined how to help them. She emphasized wanting “them to be creative" instead of following her more "experienced" approach to writing code. In many ways, this stance carried forward the self-directed approach to instruction that Alexys described receiving as a new coder: "they just sort of taught me how to like check over my work," instead of directly fixing the problem. Alexys similarly described her mentoring as not wanting to be "the leader," which to her, meant that she did not want to be "bossy" or controlling, something she did not see as part of her identity: "that's just not me." Alexys also discussed the value of being creative in her *own* coding and art practice, a theme echoed in an abstract art piece (Figure 6) from the first summer workshop (which she drew to represent "Creativity"), in her haiku about the avatar she coded in the p5 platform ("p5 more freedom / Artistic me takes over / Rosey is just made"), and in her artwork modeled after the Dear Data project (Figure 7) (in which she tracked herself making twice as many creative expressions as uncreative expressions). In short, Alexys saw herself as someone who enacted freedom, creativity, and artistic expression during coding, and as someone who practiced these same values in her mentoring.

Figure 6. Alexys's abstract artwork about "creativity"

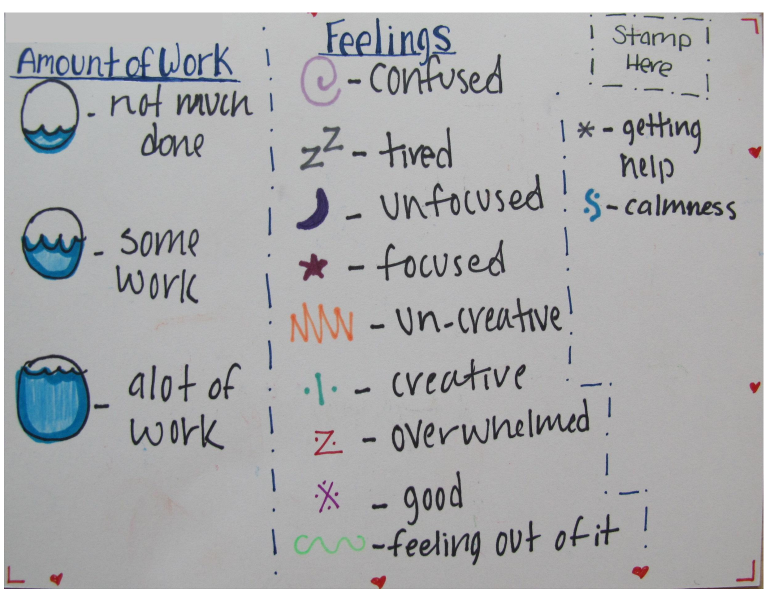

Figure 7. Alexys's data visualization artwork

In summary, Alexys described feelings of stress, freaking out, and nervousness shifting over time toward calmness, "just kind of go[ing] with it," and confidence. She explicitly credited this emotional shift to growth in her problem solving practice, which she said started as an "ask for help" approach and a sense that "you don't know what to do," and became one of really "knowing how to debug," which involved techniques that Alexys unpacked in considerable detail. For Alexys, the "whole point of coding" was wrapped up with debugging problems. By contemplating her shifting feelings and problem solving practices in our designed reflection spaces over time as she gained experience coding and debugg\ng, Alexys positioned herself as a mentor who had knowledge and wanted to help others, especially in the creative way she defined herself and wanted others to experience coding. By composing heterogenous classes of novice and more expert coders, our design opened the possibility for Alexys to articulate how others socially positioned her as a mentor. Sustained engagement over time and consistent checking in through activities like journaling and art making supported Alexys's positive perception of herself as a creative and self-assured coder who could practice cultivating those same values in her mentoring of newcomers.

Kat

Kat was a rising 8th grader during the summer of 2018. She referred to the non-profit across two separate works of art as her "second home" and where "my second family will always be." As her instructors, we took Kat to be someone who was very thoughtful about her coding and artwork; she was eager to learn and explore the bounds of her own creativity. She liked to go deep with her art, using symbolism and metaphor to express meaning. As an example of one work of coding-inspired art she made during the second summer workshop (Figure 8), she wrote in an artist statement:

Figure 8. Kat's abstract artwork

The fires of h, e, double hockey sticks
burns the soul.
The blue waves of the ocean heal
the sand of heat.
Fire, water, each come together to
make a happy medium
They create a world of peace and
destruction.
Not perfect, but it's ours

She explained during in-class storytelling with her peers that in this work of art she was trying to show with the "fire, you know, anger, frustration, all the negative emotions," and juxtaposing that with "almost like a beach, like calm, cool, you know the positive emotions." In the middle, she explained that the "purple here is when they mix together sort of yin and yang." Across art projects, Kat explored a wide range of emotion: "happiness," "confusion," "worry," "frustration," and "confidence," and marveled that "a little computer makes you feel every one of these things."

In her reflections on her problem solving experiences, Kat highlighted the centrality of frustration related to her debugging process as well as a progression of different emotions, including building confidence over time as she gained more debugging experience. She discussed the value of learning from failure to fix mistakes in the future.

Fixing mistakes

In Kat's reflections, she framed debugging as what "we use...to fix our code" and elaborated that failure is something that helped her grow as a coder, so that "the next time you know what you're doing" and can "climb the ladder." Alternatively, with some bugs, Kat noted that she nimbly adjusted her goals. Instead of fixing the bug, Kat would "change how you want it to look later on," and then drew a parallel with how in art "you can always change it to make it look like what you want it to look like." When asked how these bugs got into her code, Kat lightheartedly discussed a classroom inside joke about the *curse of Sam,* an instructor whom the students accused of magically planting bugs in their code. Kat then more seriously addressed how her mistakes derived from "not paying attention" and from how "you're just going really quickly so you didn't really notice it." Indeed, Kat saw this as a general part of who she was: "I probably just didn't notice it because I'm not very good at noticing a lot of things, like very small details I'm not very good at noticing."

Beyond taking ownership over the causes of her bugs, Kat unpacked an array of tools in her debugging kit, describing how she would "experiment," use the "stepper tool," "rewrite the code," "explain the code to someone," use "notes in our journals" to "help with the code," "ask my friends," "find error messages...and read through them," and as a last resort, "ask one of the teachers." Perhaps not surprisingly, alongside this array of debugging strategies, Kat framed persistence as a central goal. She repeatedly wrote in her coding journal about grit as her debugging goal, which she described as an approach "to keep moving forward and to keep trying." In a letter to future coders, she explained: "I'll act normal, and feel like quitting. Yet I didn't quit and I worked hard." After exhausting all of her strategies, Kat talked about soliciting help from an instructor. After a hint, she would "figure it out almost instantly," noting that she was “happy when I fix a code.” In moving forward, Kat described wanting to write down more notes to document how she fixed code to guide her problem solving process.

In all, Kat described a robust debugging practice: persisting to fix a bug or simply revising her goal, taking responsibility for bugs in her code, implementing a wide range of debugging strategies, and setting a goal of generalizing her approach to debugging. Kat’s reflections on her problem solving process and emotions connected to that process were future-oriented and focused on both the kind of coder/problem solver she thought she was and the kind of coder/problem solver she hoped to be.

Frustration building and building confidence

How did Kat respond emotionally to these stretches of debugging? Across two years, Kat consistently described wavering between confidence and frustration, what she captured precisely as “the cool yet worry of my code.” In particular, Kat talked about how she would get frustrated with her code (“#Frustration, towards a word; “#Worry, when I get an error”), especially when she felt that she had persisted within a problem solving process over a long stretch of time. This frustration was even more present if she was still clueless as to how to fix a bug she encountered: “it was very frustrating and it was very difficult because a lot of us didn’t know what to do” (first summer workshop) and “sometimes I will feel frustrated because there are times when I don’t know what’s going wrong” (second summer workshop). Like Lorena, Kat first looked to and depended on herself to understand how to approach the problems she faced when coding. She explained, “If I’m kind of like on the code for a very long time and I don’t know what I’m doing and I try fixing it a million times then I start getting a little frustrated and that’s when I ask my friends or teacher." In one of her art pieces, Kat poetically described lowering “the brightness on my screen while coding. It almost seemed as if my code disappeared…,” which Kat called “electric night.” Kat wrote about how she curbed some moments of frustration: “I view my electric night every now-and-then to have peace and be calm.”

Kat’s frustration stemmed from whether or not she considered herself confident enough to solve a problem on her own. She elaborated, “A lot of the times it could get frustrating but a lot of the times I know what I’m doing and I understand the problem." In this reflection, Kat explained that frustration was not part of her emotional experience when she could understand the problem or fix the bug, what she later called “#Confidence, once it’s over.” Kat also described emotions that followed frustration: “frustration and then afterwards there’s like curious and then there’s happiness.” In the second summer workshop, Kat made a strong statement about herself with respect to confidence: “I can get very confident very easily.” In her Dear Data piece from the second summer, she recorded herself being confident considerably more often than she recorded herself being frustrated during debugging. And in some interview responses, she alluded exclusively to confidence: “When I debugged code I would always get very confident like maybe I could do this now and I can do that, it would help with confidence….the more you do it, the more confident you get, it kind of helps.”

Kat explained that when she had an understanding of a coding problem, she experienced increased feelings of confidence in contrast to the frustration she felt when she did not know how to fix her bug. We take this to mean that Kat did not necessarily need to arrive at a solution, she just needed to understand the problem to gain confidence. Kat discussed a progression of emotions that led her to understand if she knew how to solve a problem and therefore, if she had confidence. The progression of emotions included a bit of frustration, “curiousness,” happiness, and “more curiousness” before she figured it out. She explained that the more she debugged, the more her confidence grew.

In summary, Kat talked about a range of emotions experienced during problem solving, including frustration with bugs as well as curiosity and happiness. Her emotions informed decisions about problem solving, including whether she would ask for help, persevere, or take a break. She explained how past mistakes helped her build confidence over time and envisioned a future self as growing even more confident. Her stories described temporal trajectories that interleaved emotion and problem solving, for example, when Kat remarked on how she showed no emotion on the surface, felt like quitting on the inside, and yet continued working toward a solution, exhausting all of her debugging strategies, asking for help, and then seeing the solution and feeling happy. In another account, Kat describes debugging without arriving at a candidate fix, feeling frustration, and then darkening her screen to take a break. For Kat, these were trajectories through moments of failure that strung particular emotions together with particular problem solving experiences, and allowed at times for a strong sense of confidence and identification as someone who is persistent. Art making and storytelling opportunities allowed Kat to surface how she saw herself as a coder, namely how she saw herself changing as a coder, both over short and long time scales, and with respect to her past, present, and future progress. These activities afforded Kat a context to precisely narrate how her emotional experiences and problem solving approaches interacted to make coding feel a certain way at certain points in time.

Discussion

As Lorena, Alexys, and Kat discussed coding-linked emotional experiences, they talked about their problem solving experiences in relation to how they saw themselves as coders (see Table 2). More concretely, participants' ways of seeing themselves as particular kinds of coders (*slow, different from everyone else, creative, persistent, independent*) were linked to how they described their problem solving process (*overthinking, re-reading, stepping, taking breaks*) and how they reported feeling (*sucks, peace and calm, getting frustrated, confident, freaking out, lacking nervousness*). Without attending to emotion in our design, it would have been difficult to infer how participants' approaches to problem solving informed their evolving identities. By acknowledging emotion and engaging with how emotion connected with debugging processes, participants' creative reflections created a bridging space between problem solving and identity. What *feels* right, wrong, comfortable, or uncomfortable when navigating a problem forms a constitutive element of identity (Lemke, 2010). Considered in light of other facets of identity, including stories others tell about us (Sfard & Prusak, 2005) and common practices across communities (Nasir, 2002), the emotional context in which problem solving takes place becomes one more grounded way to understand how students *experience* a discipline (Sengupta et al., 2018).

As computer science educational researchers continue to call attention to the value of students' computational identities (Tissenbaum et al., 2019), in particular to foster more inclusive learning spaces, we would argue that the portraits presented in this paper nudge teachers and researchers to attend to identity at the intersection of how learners see themselves solving problems *and* how they feel about their approach. We complicate the notion of confidence, something we know to be particularly important for girls identifying with computer science (e.g., Friend, 2015; Hur et al., 2017), by showing how it changes over time relative to problem solving skill. Additionally, through our interest in creating a productive culture around failure, we positioned identity as something that requires social support (Friend, 2015). Viewed in light of prior work capturing substantive shifts in identity over time (Heyd-Metzuyanim, 2015), we argue that these portraits reaffirm that educators should be open to seeing identity as malleable, notice when identity changes take place, and question how other facets of experience, whether problem solving approaches, emotions, or stories, contribute along the way (e.g., Fields & Enyedy, 2013).

Table 2. Overview of girls' problem solving approaches, related emotions, and identifying statements

Girl	Problem solving approach and emotions	How she defines herself
Lorena	- Overthinking problems *sucks* and is a *bad habit* but also helps to see the solution - Re-reading code - Working independently even while *freaking out* - Getting stuck, feeling *embarrassed*, and not moving on from the bug...over time becomes getting stuck, feeling *calm*, following debugging strategies, and moving along	Someone who wants to feel in control, is meticulous, works independently, moves slowly, overthinks the solution, has a hard time getting to the solution, approaches problems differently from "everyone else," and has future aspirations of less stress and thinking of easier solutions
Alexys	- Not knowing how to debug, *freaking out* and *stressing* within an *explosion*, and relying on help...over time becomes knowing how to debug (stepper, check spelling, console, re-read, double check, revise, look at prior work) and *lacking nervousness, going with it*, and feeling *confident*	Someone who is a mentor who knows what she is doing with coding and *feels good* about that; not a bossy kind of person; someone who lets others be creative; a creative person herself; a confident person in relation to coding ability and knowledge
Kat	- Learning from past failures - Changing the goal to avoid the bug - Using a wide range of debugging strategies (e.g., experiment, step, explain) before asking for help from an instructor - *Feeling like quitting* and *getting frustrated* but *acting normal* when facing a bug she does not understand, but choosing to persevere, feeling *curiosity* and eventually feeling *confidence* once it's over - Taking breaks to find *peace and calm* when faced with prolonged debugging	Someone who goes quickly, doesn't always pay attention, is not good at noticing details, perseveres through tough bugs, starts by working independently, grows confidence over time as she gains experience, and can easily get confident.

Implications for and limitations for the design of more inclusive learning environments

The finding that problem solving, emotion, and identity are intertwined points to a number of key implications as illustrated in our three portraits. We highlight two here. First, we argue that creating an inclusive learning environment requires checking in with learners to track not only whether learning is taking place, but also to understand how they are feeling when engaged in practice. In the case of our portraits, we asked all three girls to consider the emotions involved in their practice through a variety of reflective practices such as journaling, art making, and interviews with instructors. Alexys, for example, described feelings of stress and nervousness contrasting with her eventual experiences of being calm and gaining confidence. Similarly, both Lorena and Kat deeply engaged with prompts to reflect on emotions, acknowledging that they were often mixed up, overlapping, and shifted over time. In computer programming contexts, in which moments of failure are well-recognized as foundational, common, and challenging (McCauley et al., 2008), including inviting of complex emotion (Bosch et al., 2013; Dahn et al., 2020; Kinnunen & Simon, 2010), we would be asking newcomers to process recurring failures in isolation unless we attend carefully to and support their experiences. Indeed, our approach to supporting learners *during coding and debugging* often emphasized tool-driven debugging strategies (DeLiema et al., 2020). In doing so, we inadvertently separated reflection on problem solving from reflection on other aspects of coding experience. Bridging this gap more directly during coding instruction is an opportunity for future research.

Second, it is important that problem solving, emotion, and identity as constructs are not isolated in future designs as the portraits showed us that all three are deeply connected. For example, we saw that Lorena's description of herself as someone who saw things differently was connected to the scattered and embarrassed emotions she described as she compared herself to others during her coding practice. Similarly, Kat's mixed feelings of frustration, confusion, and happiness were related to precise moments in her problem solving process and contributed to her growing confidence. Thus, if we want to support particular identities, we should make sure students are also supported with their problem solving practice as well as take inventory of how they feel along the way. And by extension, if we want to make sure students have strong problem solving practices, then we should also check in on their identities as coders and their emotions. In short, we want to make sure we understand our students and how they are changing over time along these three dimensions if we want to responsively support their growth as coders. Importantly, attending to these three dimensions in ways that can make a difference for how learners see themselves must be an embedded part of the learning design *throughout* the experience and cannot simply be a summative afterthought. That is, artificially asking students how something felt at the end of a coding workshop will not likely result in a transformative understanding of their identities or problem solving approaches. Though we did try to embed reflection spaces into our curriculum design throughout, in future research we might try to even more explicitly integrate reflection on emotion and identity during students' coding practice.

How can we embed formative dialogue around emotion and identity in computer science learning contexts, and what purpose would this serve? Our design approach takes advantage of the open-ended, reflective space of art making to foreground stories about success and failure. Without a doubt, this approach is not the only one available to foster reflection. More broadly, in envisioning new classroom practices aligned with our designs, we suggest building on research recommending that interventions place youth in a position of power (e.g., Yeager et al., 2018). Foster (2014) richly describes how mathematics teaching that reflects a *pastoral* approach to instruction foregrounds "warm empathetic listening and minimal intervention to support people in solving their own problems and developing increased autonomy" (p. 147). Foster notes that teaching along these lines involves instructors giving considerable time to listening to their students, remaining open to multiple possibilities, reflecting with colleagues, and feeling uncertain. Positioning students as authoritative drivers of action is perhaps particularly important to dialogue around emotion. As Jaber & Hammer (2016) note, instructors should "accommodate a range of norms regarding emotional expression" and expect that students will "vary significantly with respect to how they experience, express, and regulate how they are feeling, both across and within cultures" (p. 191). Efforts to respond to students' heterogeneous reflections in this way might align well with an institutional "culture of care" in schools (Cooper & Chickwe, 2012), in which the "caring actions of school officials must be motivated by student need, not an adult's desire to demonstrate sympathy, empathy, or compassion" (p. 242), and should be accompanied by positive peer relationships and high expectations. More work is needed to continue to describe the ways that instructors foreground learners' authority within classrooms that surface nuanced reflections on emotion, problem solving, and identity. In addition, perhaps future research can consider how key elements of instructional and pedagogical design focused on learner agency and distribution of authority might further support how we theorize about computer science identity (see Cobb et al., 2009 for similar discussion of identities in mathematics classrooms).

We acknowledge the limitations of generalizing from our unique dataset, in which all focus participants had multiple years of programming experience. Unfortunately, although they may spark interest, most drop-in summer camps may not be enough to make a difference in girls' identifications with computer science due to their short duration (e.g., Hur et al., 2017). School contexts also pose challenges (e.g., limited time for certain disciplines, student-teacher ratios) to the translational work needed to create spaces that foreground emotion and identity. One way comprehensive public schools might look to support learners could be through the creation of smaller learning communities built around shared interests in computer programming (Edward et al., 2019).

Our study also presents limitations in terms of how learners' cultures might be explicitly incorporated into design to support identity development. Both formal and informal classrooms spaces are figured worlds (Holland et al., 1998) with particular sociodisciplinary values and norms. In our case, while participants engaged in peer mentorship, our curriculum emphasized developing confidence and skill as an *independent* problem solver during debugging (e.g., DeLiema et al., 2020). The discourse, task, and material structures that were informed by this value in our design may have aligned or clashed with learners' familiar, culturally based practices outside the classroom (e.g., Lee, 1995). For example, some research suggests that learners belonging to Indigenous communities value collaboration centered on ensemble and unified engagement rather than the joining together of separate ideas (Mejía-Arauz et al., 2018). This raises the possibility that participants in our learning community encountered friction moving between distinct cultural activity systems (Engeström, 2001). Understanding learners' cultural viewpoints and how they approached sense-making would likely have afforded an even stronger opportunity to support our students. Future work could explore how to build from our focus on emotion during problem solving to design for *rightful presence* (Calabrese Barton, & Tan, 2019) by making visible youth experiences with marginalization and then using those experiences to drive their engagement with consequential learning (Jurow et al., 2016). We might also consider how integrating personal narratives within curriculum could surface girls' understandings of how their race, ethnicity, and gender intersect with their computer science identities (Pinkard et al., 2017). Explicitly designing from a critical, social justice-oriented perspective coupled with the approach of attending to emotion during the problem solving process strikes us as an important opportunity for future research.

Here we have shown how strong girl coders make sense of their debugging experiences through reflections on emotion and identity during problem solving. The portraits drew attention to interconnected dynamics between these three constructs, substantive shifts in students' practices and experiences over time, and implications for the design of computer programming learning environments. Our work suggests that designing learning environments that attend to what learners are thinking and feeling when they solve challenging problems creates space for a computer science identity to develop with implications for both present and future practice.

Acknowledgments

This work was supported by the National Science Foundation under Grant Nos. 1612770, 1607742, and 1612660.

References

Ashcraft, C., Eger, E., & Friend, M. (2012). *Girls in IT: The facts. National Center for Women and Information Technology.*

Bandura, A. (1982). Self-efficacy mechanism in human agency. *American Psychologist, 37*(2), 122–147. https://doi.org/10.1037/0003-066X.37.2.122

Bennedsen, J., & Caspersen, M. E. (2008). Optimists have more fun, but do they learn better? On the influence of emotional and social factors on learning introductory computer science. *Computer Science Education, 18*(1), 1–16. https://doi.org/10.1080/08993400701791133

Bosch, N., D'Mello, S., & Mills, C. (2013). What emotions do novices experience during their first computer programming learning session? In H. C. Lane, K. Yacef, J. Mostow, & P. Pavlik (Eds.), *Artificial intelligence in education* (pp. 11–20). Springer.

Brickhouse, N. W., & Potter, J. T. (2001). Young women's scientific identity formation in an urban context. *Journal of Research in Science Teaching, 38*(8), 965–980. https://doi.org/10.1002/tea.1041

Brown, A. L. (1992). Design experiments: Theoretical and methodological challenges in creating complex interventions in classroom settings. *Journal of the Learning Sciences, 2*(2), 141–178. https://doi.org/10.1207/s15327809jls0202_2

Buchholz, B., Shively, K., Peppler, K., & Wohlwend, K. (2014). Hands on, hands off: Gendered access in sewing and electronics practices. *Mind, Culture, and Activity, 21*(4), 1–20. https://doi.org/10.1080/ 10749039.2014.939762

Calabrese Barton, A., Kang, H., Tan, E., O'Neill, T. B., Bautista-Guerra, J., & Brecklin, C. (2013). Crafting a future in science: Tracing middle school girls' identity work over time and space. *American Educational Research Journal, 50*(1), 37–75. https://doi.org/10.3102/0002831212458142

Calabrese Barton, A., & Tan, E. (2019). Designing for rightful presence in STEM: The role of making present practices. *Journal of the Learning Sciences, 28*(4–5), 1–43. https://doi.org/10.1080/ 10508406.2019.1591411

Cobb, P., Gresalfi, M., & Hodge, L. L. (2009). An interpretive scheme for analyzing the identities that students develop in mathematics classrooms. *Journal for Research in Mathematics Education, 40*(1), 40–68.

Collins, A. (1992). Toward a design science of education. In E. Scanlon & T. O'Shea (Eds.), *New directions in educational technology* (pp. 15–22). Springer-Verlag.

Cooper, R., & Chickwe, M. (2012). Building bridges between urban schools and urban commu- nities: Creating an institutional culture of care in schools. In C. Boske (Ed.), *Educational leadership: Building bridges among ideas, schools and nations* (pp. 233–239). Information Age Publishing.

Cope, V., Jones, B., & Hendricks, J. (2016). Why nurses chose to remain in the workforce: Portraits of resilience. *Collegian, 23*(1), 87–95. https://doi.org/10.1016/j.colegn.2014.12.001

D'Mello, S., & Graesser, A. (2012). Dynamics of affective states during complex learning. *Learning and Instruction, 22*(2), 145–157. https://doi.org/10.1016/j.learninstruc.2011.10.001

Dahn, M., DeLiema., D., & Enyedy, N. (2020). Art as a point of departure for storytelling about the experience of learning to code. *Teachers College Record, 122*(8).

Dasgupta, N., & Stout, J. G. (2014). Girls and women in science, technology, engineering, and mathematics. *Policy Insights from the Behavioral and Brain Sciences, 1*(1), 21–29. https://doi.org/ 10.1177/2372732214549471

DeBellis, V. A., & Goldin, G. A. (2006). Affect and meta-affect in mathematical problem solving: A representational perspective. *Educational Studies in Mathematics, 63*(2), 131–147. https://doi.org/10.1007/s10649-006-9026-4

Dekhane, S., & Napier, N. (2017). Impact of participation vs. non: Participation in a programming boot camp (PBC) on women in computing. *Journal of Computing Sciences in Colleges, 33*(2), 245–252.

DeLiema, D., Dahn, M., Flood, V. J., Asuncion, A., Abrahamson, D., Enyedy, N., & Steen, F. F. (2020). Debugging as a context for collaborative reflection on problem-solving processes. In E. Manolo (Ed.), *Deeper learning, communicative competence, and critical thinking: Innovative, research-based strategies for development in 21st century classrooms* (pp. 209–228). Routledge.

Denzin, N. K. (1978). *Sociological methods: A sourcebook* (2nd ed.). McGraw-Hill.

Edward, C., Fletcher, N. W., & Hernandez-Gantes, V. M. (2019). The high school academy as a laboratory of equity, inclusion, and safety. *Computer Science Education, 29*(4), 382–406. https://doi.org/10.1080/08993408.2019.1616457

Engeström, Y. (2001). Expansive learning at work: Toward an activity theoretical reconceptualization. *Journal of Education and Work, 14*(1), 133–156. https://doi.org/10.1080/13639080020028747

Fields, D., & Enyedy, N. (2013). Picking up the mantle of "expert": Assigned roles, assertion of identity, and peer recognition within a programming class. *Mind, Culture, and Activity, 20*(2), 113–131. https://doi.org/10.1080/10749039.2012.691199

Foster, C. (2014). Minimal interventions in the teaching of mathematics. *European Journal of Science and Mathematics Education, 2*(3), 147–154.

Friend, M. (2015). Middle school girls' envisioned future in computing. *Computer Science Education, 25*(2), 152–173. https://doi.org/10.1080/08993408.2015.1033128

Frieze, C., Quesenberry, J. L., Kemp, E., & Velázquez, A. (2012). Diversity or difference? New research supports the case for a cultural perspective on women in computing. *Journal of Science Education and Technology, 21*(4), 423–439. https://doi.org/10.1007/s10956-011-9335-y

Glaser, B. G. (1965). The constant comparative method of qualitative analysis. *Social Problems, 12*(4), 436–445. https://doi.org/10.2307/798843

Goodwin, M. H., Cekaite, A., & Goodwin, C. (2012). Emotion as stance. In A. Peräkylä & M.-L. Sorjonen (Eds.), *Emotion in interaction* (pp. 16–41). Oxford University Press.

Hand, V., & Gresalfi, M. (2015). The joint accomplishment of identity. *Educational Psychologist, 50*(3), 190–203. https://doi.org/10.1080/00461520.2015.1075401

Herman, D. (2009). *Basic elements of narrative*. John Wiley & Sons.

Heyd-Metzuyanim, E. (2015). Vicious cycles of identifying and mathematizing: A case study of the development of mathematical failure. *Journal of the Learning Sciences, 24*(4), 504–549. https://doi. org/10.1080/10508406.2014.999270

Holland, D., Lachicotte, W., Jr., Skinner, D., & Cain, C. (1998). *Identity and agency in cultural worlds*. Harvard University Press.

Hur, J., Andrzejewski, C., & Marghitu, D. (2017). Girls and computer science: Experiences, perceptions, and career aspirations. *Computer Science Education, 27*(2), 100–120. https://doi.org/10.1080/08993408.2017.1376385

Jaber, L. Z., & Hammer, D. (2016). Learning to feel like a scientist. *Science Education, 100*(2), 189–220. https://doi.org/10.1002/sce.21202

Jurow, A. S., Teeters, L., Shea, M., & Van Steenis, E. (2016). Extending the consequentiality of "invisible work" in the food justice movement. *Cognition and Instruction, 34*(3), 210–221. https://doi.org/10. 1080/07370008.2016.1172833

Kapur, M. (2008). Productive failure. *Cognition and Instruction, 26*(3), 379–424. https://doi.org/10.1080/07370000802212669

Kinnunen, P., & Simon, B. (2010). Experiencing programming assignments in CS1: The emotional toll. In *Proceedings of the Sixth International Workshop on Computing Education Research*. (pp. 77–85). Aarhus, Denmark: ACM.

Kinnunen, P., & Simon, B. (2012). Phenomenography and grounded theory as research methods in computing education research field. *Computer Science Education, 22*(2), 199–218. https://doi.org/ 10.1080/08993408.2012.692928

Koschmann, T., Kuutti, K., & Hickman, L. (1998). The concept of breakdown in Heidegger, Leont'ev, and Dewey and its implications for education. *Mind, Culture, and Activity, 5*(1), 25–41. https://doi. org/10.1207/s15327884mca0501_3

Lawrence-Lightfoot, S. (1986). On goodness in schools: Themes of empowerment. *Peabody Journal of Education, 63*(3), 9–28. https://doi.org/10.1080/01619568609538522

Lawrence-Lightfoot, S. (2016). Portraiture methodology: Blending art and science. *LEARNing Landscapes, 9*(2), 19–27. https://doi.org/10.36510/learnland.v9i2.760

Lawrence-Lightfoot, S., & Davis, J. (1997). The art and science of portraiture. Jossey-Bass.

Lee, C. (1995). Signifying as a scaffold for literary interpretation. *Journal of Black Psychology, 21*(4), 357–381. https://doi.org/10.1177/00957984950214005

Lehman, K. J., Sax, L. J., & Zimmerman, H. B. (2016). Women planning to major in computer science: Who are they and what makes them unique? *Computer Science Education, 26*(4), 277–298. https:// doi.org/10.1080/08993408.2016.1271536

Lemke, J. L. (2010). Affect, identity, and representation. In K. Gomez, L. Lyons, & J. Radinsky (Eds.), *Proceedings of the ninth international conference of the learning sciences*. Chicago, IL: International Society of the Learning Sciences.

Lerner, J. S., Li, Y., Valdesolo, P., & Kassam, K. S. (2015). Emotion and decision making. *Annual Review of Psychology, 66*(1), 799–823. https://doi.org/10.1146/annurev-psych-010213-115043

Lewis, C. M., Anderson, R. E., & Yasuhara, K. (2016). I don't code all day: Fitting in computer science when the stereotypes don't fit. In *Proceedings of the 2016 ACM conference on international computing education research* (pp. 23–32). Melbourne, Australia: ACM.

Lui, D., Kafai, Y., Litts, B., Walker, J., & Widman, S. (2020). Pair physical computing: High school students' practices and perceptions of collaborative coding and crafting with electronic textiles. *Computer Science Education, 30*(1), 72–101. https://doi.org/10.1080/08993408.2019. 1682378

Lyon, L. A., & Green, E. (2020). Women in coding boot camps: An alternative pathway to computing jobs. *Computer Science Education, 30*(1), 102–123. https://doi.org/10.1080/08993408.2019. 1682379

Margolis, J., & Fisher, A. (2003). *Unlocking the clubhouse: Women in computing*. MIT Press.

Margolis, J., Ryoo, J. J., Sandoval, C. D., Lee, C., Goode, J., & Chapman, G. (2012). Beyond access: Broadening participation in high school computer science. *ACM Inroads, 3*(4), 72–78. https://doi. org/10.1145/2381083.2381102

Master, A., Cheryan, S., & Meltzoff, A. N. (2016). Computing whether she belongs: Stereotypes undermine girls' interest and sense of belonging in computer science. *Journal of Educational Psychology, 108*(3), 424–437. https://doi.org/10.1037/edu0000061

McCauley, R., Fitzgerald, S., Lewandowski, G., Murphy, L., Simon, B., Thomas, L., & Zander, C. (2008). Debugging: A review of the literature from an educational perspective. *Computer Science Education, 18*(2), 67–92. https://doi.org/10.1080/08993400802114581

Mejía-Arauz, R., Rogoff, B., & Paradise, R. (2005). Cultural variation in children's observation during a demonstration. *International Journal of Behavioral Development, 29*(4), 282–291. https://doi.org/10.1177/01650250544000062

Nasir, N. (2002). Identity, goals, and learning: Mathematics in cultural practice. *Mathematical Thinking and Learning, 4*(2–3), 213–247. https://doi.org/10.1207/S15327833MTL04023_6

Nasir, N., & Hand, V. (2008). From the court to the classroom: Opportunities for engagement, learning, and identity in basketball and classroom mathematics. *Journal of the Learning Sciences, 17*(2), 143–179. https://doi.org/10.1080/10508400801986108

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

National Academies. (2007). *Beyond bias and barriers: Fulfilling the potential of women in academic science and engineering.*

National Center for Education Statistics. (2011). The nation's report card. U.S. Department of Education.

Oatley, K. (1987). Cognitive science and the understanding of emotions. *Cognition & Emotion, 1*(3), 209–216. https://doi.org/10.1080/02699938708408048

Palinscar, A. S., & Brown, A. L. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition and Instruction, 1*(2), 117–175. https://doi.org/10.1207/s1532690xci0102_1

Pantic, K., & Clarke-Midura, J. (2019). Factors that influence retention of women in the computer science major: A systematic literature review. *Journal of Women and Minorities in Science and Engineering, 25*(2), 119–145. https://doi.org/10.1615/JWomenMinorScienEng.2019024384

Peppler, K. (2016). ReMaking arts education through physical computing. In K. Peppler, E. Halverson, & Y. Kafai (Eds.), *Makeology: Makers as learners* (Vol. 2, pp. 206–226). Routledge.

Pinkard, N., Erete, S., Martin, C. K., & de Royston, M. M. (2017). Digital youth Divas: Exploring narrative-driven curriculum to spark middle school girls' interest in computational activities. *Journal of the Learning Sciences, 26*(3), 477–516. https://doi.org/10.1080/10508406.2017.1307199

Ramirez, G., Gunderson, E. A., Levine, S. C., & Beilock, S. L. (2013). Math anxiety, working memory, and math achievement in early elementary school. *Journal of Cognition and Development, 14*(2), 187–202. https://doi.org/10.1080/15248372.2012.664593

Richardson, L. (1997). *Fields of play: Constructing an academic life*. Rutgers University Press.

Rodriguez, S. L., & Lehman, K. (2017). Developing the next generation of diverse computer scientists: The need for enhanced, intersectional computing identity theory. *Computer Science Education*, 27(3–4), 229–247. https://doi.org/10.1080/08993408.2018.1457899

Sandoval, W. A. (2012). Situating epistemological development. In J. van Aalst, K. Thompson, M. J. Jacobson & P. Reimann (Eds.), *The future of learning: Proceedings of the 10th International*

Conference of the Learning Sciences (Vol. 1, pp. 347–354). Sydney, Australia: International Society of the Learning Sciences.

Sengupta, P., Dickes, A., & Farris, A. (2018). Toward a phenomenology of computational thinking in STEM education. In M. S. Khine (Ed.), *Computational thinking in STEM discipline: Foundations and research highlights* (pp. 49–72). Springer.

Sfard, A., & Prusak, A. (2005). Telling identities: In search of an analytic tool for investigating learning as a culturally shaped activity. *Educational Researcher, 34*(4), 14–22. https://doi.org/10.3102/0013189X034004014

Shah, N., & Lewis, C. M. (2019). Amplifying and attenuating inequity in collaborative learning: Toward an analytical framework. *Cognition and Instruction, 37*(4), 423–452. https://doi.org/10.1080/07370008.2019.1631825

Tissenbaum, M., Sheldon, J., & Abelson, H. (2019). From computational thinking to computational action. *Communications of the ACM, 62*(3), 34–36. https://doi.org/10.1145/3265747

Tonso, K. L. (2006). Student engineers and engineer identity: Campus engineer identities as figured world. *Cultural Studies of Science Education, 1*(2), 273–307. https://doi.org/10.1007/s11422-005-9009-2

Tracy, S. J. (2010). Qualitative quality: Eight "big-tent" criteria for excellent qualitative research. *Qualitative Inquiry, 16*(10), 837–851. https://doi.org/10.1177/1077800410383121

Turkle, S., & Papert, S. (1992). Epistemological pluralism and the revaluation of the concrete. *Journal of Mathematical Behavior, 11*(1), 3–33.

Varma, R. (2010). Why so few women enroll in computing? Gender and ethnic differences in students' perception. *Computer Science Education, 20*(4), 301–316. https://doi.org/10.1080/08993408.2010.527697

Weiner, B. (1985). An attributional theory of achievement motivation and emotion. *Psychological Review, 92*(4), 548–573. https://doi.org/10.1037/0033-295X.92.4.548

Wiggins, S. (2016). *Discursive psychology: Theory, method and applications*. Sage.

Wong, B. (2016). I'm good, but not that good': Digitally-skilled young people's identity in computing. *Computer Science Education, 26*(4), 299–317. https://doi.org/10.1080/08993408.2017.1292604

Wortham, S. (2006). *Learning identity: The joint emergence of social identification and academic learning*. Cambridge University Press.

Yeager, D. S., Dahl, R. E., & Dweck, C. S. (2018). Why interventions to influence adolescent behavior often fail but could succeed. *Perspectives on Psychological Science, 13*(1), 101–122. https://doi.org/ 10.1177/1745691617722620

Zeller, A. (2009). *Why programs fail: A guide to systematic debugging*. Elsevier.