

©Copyright 2026

jayne everson

computing is for everybody:
pedagogy for joyful secondary computing

jayne everson

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2026

Reading Committee:

Amy J. Ko, Chair

R. Benjamin Shapiro

Nassim Parvin

Maya Cakmak

Program Authorized to Offer Degree:
Paul G. Allen School of Computer Science and Engineering

University of Washington

Abstract

computing is for everybody:
pedagogy for joyful secondary computing

jayne everson

Chair of the Supervisory Committee:
Amy J. Ko
Information School

Computing has never had a broader impact on the world: it surrounds us and shapes our daily lives. In contrast to its broad influence, a select few are designing and building it. To have a democratic future, we need all voices represented in technology's creation. We need to imagine new inclusive futures for secondary computing education: one where all students are excited and engaged in constructing their own knowledge, and building the future.

In this work, I examine joyful pedagogies in secondary computing education. Using the methods of interviews and speculative design, I characterize the educational experiences of students and teachers and how they manifest. Informed by these insights I implemented them in a computational embroidery class. These findings offer us an understanding of how to create more joyful computing education, and prove my thesis: feminist pedagogies encourage joyful secondary computing.

TABLE OF CONTENTS

	Page
List of Figures	iii
List of Tables	v
Chapter 1: Introduction: Building a future for everybody	3
Chapter 2: Background and Theoretical Foundations	12
2.1 Grounded in Joyful Learning	12
2.2 What is joyful learning?	13
2.3 Miserable Learning (Hegemonic defaults in STEM and computing)	15
2.4 Joy in Pedagogy	16
2.5 What could joyful CS pedagogy be?	29
Chapter 3: The state of CS: Deconstructing Hegemonic Defaults in Computing by Interrogating the concept of Rigor	31
3.1 Importance of Understanding Perceptions of Rigor	31
3.2 Existing Conceptions of Rigor	33
3.3 Methods for Investigating Rigor in CS education	38
3.4 Conceptions of Rigor in CS	43
3.5 Discussion	63
3.6 Conclusion	70
Chapter 4: Student Dreams of Joyful Learning: Dreaming of Difference	71
4.1 Why do we need to ask students what they want?	71
4.2 Prior Work	73
4.3 Methods for Dreaming of CS Classrooms	77
4.4 Student's Dreams of CS	86
4.5 Discussion	97

Chapter 5:	Joyful Environments: Students' Experiences of Joyful Secondary Computing	102
5.1	What is joy in computing education?	102
5.2	What makes computing joyful for students?	104
5.3	Related work	105
5.4	Methods	108
5.5	Elements of Joyful Computing Experiences	113
5.6	Discussion	130
Chapter 6:	Teachers' Experiences of Joyful Learning: Postcards from the Field	135
6.1	What is joyful for teachers?	135
6.2	Methods	136
6.3	Joyful K-12 Computing	140
6.4	Discussion	150
Chapter 7:	Joyful Content: An Experiment with Computational Embroidery	152
7.1	Testing feminist pedagogies with a feminine medium	153
7.2	Related Work	156
7.3	Co-Constructing a Computational Embroidery Learning Space	158
7.4	Methods	164
7.5	How FCE shapes learning and teaching	166
7.6	Discussion	188
Chapter 8:	Where do we build from here?	195
8.1	What is joy in computing?	195
8.2	What we need to build: future work	200
8.3	Recommendation for joyful pedagogies and practices	204
8.4	There is no conclusion	208

LIST OF FIGURES

Figure Number	Page
2.1 Theories and frameworks overlap and many theories carry the same pedagogies. The goal of this study is to identify which pedagogies encourage joyful learning.	17
2.2 Joyful CS pedagogies are feminist pedagogies sit at the center of critical, constructionist, and feminist pedagogies. Feminism centers relationality as well as the embodied construction of knowledge, while acknowledging the systemic inequities of the world.	28
3.1 Each participant's conceptions of rigor sorted by type of participant. There is a heat map of popularity of conceptions at the end of each participant group and overall. Note that no two participants definitions match.	44
4.1 Vision boards and student analysis. On the left are some of the vision boards students created in workshop 2. On the right is the affinity diagramming board that students and an organizer constructed during workshop 3.	82
4.2 Responsibility the interdependent model of responsibility and accountability and that students described.	88
4.3 A boat filled with the students' desired attributes of a CS classroom against the current of a river full of students' CS perception.	96
5.1 The research team conceptualized the reification of a joyful classroom with the above diagram of four elements of what Teachers were doing to create these joyful environments. Each of the four elements has several different components. Not all students mentioned each of the components, but they did mention more than one component from each of the four elements.	114
6.1 Poster from the resources small group written and collected at the workshop. . .	137
6.2 Design probes distributed to teachers.	138
6.3 Postcard 3 is an example of the postcards teachers mailed in. They used the art tools to illustrate and share their thoughts and also often included a few sentences or short notes.	141
6.4 Postcard 4 shares a teacher's perception of the AHA moment that so many students find as a joyful outcome of studying CS.	144

7.1	On the left is a screenshot of a student's TurtleStitch code from Week 3. On the right is the machine embroidering the work.	154
7.2	Red holding up the finished boba tea cup – looking like she wanted it to. . . .	168
7.3	On the left is a snapshot of Chocolate Milk's TurtleStitch Corgi, On the right was the work he shared at the end of term program Banquet. The red circle highlights a stitch Chocolate Milk had discovered that day, and was intending to use to fill in the corgi's rear end after the end of the course demonstrating his fluency using both machine and hand embroidery techniques.	170
7.4	Bob showing off 'alit' put together with 'the silly mad goat' he made for his final project.	171
7.5	Ricky Bobby's work from left to right: Ricky Bobby running the embroidery machine, The middle image is the code for an Aztec pattern, Ricky Bobby's final work inspired by Mexican floral embroidery traditions.	171
7.6	A snapshot of the code snippet and an image of jellyfish's circuit and jellyfish debugging the machine while embroidering their circuit.	174
7.7	The evolution of JMGT's work- initially she started with a lotus flower, and then improved both her code, and the flower, and eventually included those flowers in her final piece along with raspberries and bees.	174
7.8	The left image is Eric's Lotus code that he worked meticulously on. On the right is Eric working on his final project telling a childhood story by combining his love of painting and the embroidery techniques he learned.	175
8.1	Joyful CS pedagogies are feminist pedagogies sit at the center of critical, constructionist, and feminist pedagogies. Feminism offers the relationality as well as the embodied construction of knowledge, while acknowledging the systemic inequities of the world.	198

LIST OF TABLES

Table Number	Page
3.1 Participants self-reported gender, race/ethnicity, and other identities with the authors, and shared the major and CS courses they had taken as students or level and courses they were teaching as instructors.	40
3.2 Conceptions of rigor participants described, organized by broader groupings, then conceptions and a description of each conception.	43
4.1 Participants' self-disclosed demographic information	78
4.2 Final codebook for <i>What CS education should be</i> from thematic analysis	85
5.1 Table of the students, joyful educators, and brief context of the environment of joyful CS	110
5.2 Codebook of Themes and the associated elements in a joyful CS experience. .	112
7.1 Students self-selected pseudonyms and self-reported demographics of age, gender identity, race and ethnicity, first-generation college student status, and any other identities they wanted researchers to know. Students could opt out of any question.	162
7.2 Weekly course breakdown with each week's topic, co-constructed assignment, and an example of student work. Students began learning about hand embroidery and the cultural traditions of embroidery. Students then learned how to program with TurtleStitch and use the embroidery machine.	165
7.3 Themes and claims.	167
8.1 Recommendations for joyful pedagogies in computing education and examples from this study	205

ACKNOWLEDGMENTS

first, for each of my students who have offered curiosity and joy and extended patience. towana for letting me do math freely, even if it took all day. karen, thank you for sharing the readings and the words to match the experience, and for setting an example of joyful learning. rob macdonald, thank you for always extending grace and trust as i tested liberatory pedagogical practices in my classes and in the design center and for pushing me to examine my assumptions. lovely ladies, stay fresh cheesebags. my labmates have made this time at uw joyful, and have written with me, and offered critique to make me a better writer. rotem, thank you for the co-writing, the care, and the nachos. nassim, thank you for a joyful classroom, and for inviting me into the salon. amy, thank you for the critical care that is love and the inordinate patience you have extended as i learn to write. nick, thank you for all the snacks and all the things. i couldn't have written this without you.

DEDICATION

for Rod and Dr. Way who made learning technical things joyful

PROLOGUE

SUBJECTIVITY AND OBJECTIVITY: POSITIONALITY

My road to teaching and studying computer science is not linear. I started as a mechanical engineer in Chicago. I've lived and taught in Central Asia, Kentucky, Boston, and Seattle. I hold professional identities of engineer, teacher, and graduate student. In each job I've held I've faced gender based discrimination. My technical expertise has been unseen, forgotten, and questioned. I've watched sexism play out in the classroom in assumptions made about students and their abilities based on how they present.

Broadly, I recognize many of those experiences as sexist. Narrowly, those experiences feel terrible and unfortunately have influenced my self-perception and my self-efficacy in all realms; technical, personal, and pedagogical. I have also been complicit in sexism. I am still (unintentionally) complicit. Dismantling my own internalized sexism will be a lifelong task. Sexism is just one form of oppression I have experienced and been complicit in.

I have multiple physical and learning disabilities. Those disabilities have informed my lived experience. I view the act of writing this document as an act of resistance.

My philosophy of education is that learning is the best and should be full of joy. I believe that each child in my classroom should have the ability to learn things that are relevant and meaningful and that they care about. They should each have their voices heard and valued, and should have the ability to make choices in the classroom. Each child brings their own experiences, personality, and strengths. It is important to approach an individual's learning from an assets-based approach, where they are encouraged and feel confident to wrestle with work and problems they create or encounter.

I also believe that learning is an embodied cognitive process, and I make sense of the world around me by creating and applying what I'm learning. I believe that theory must be carried out

in practice. This influences how I examine pedagogy.

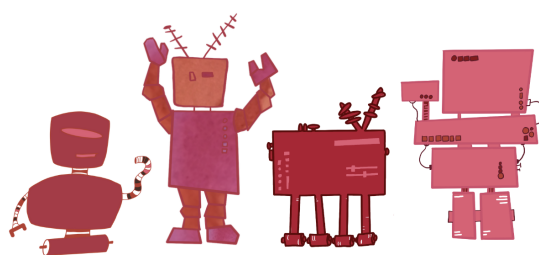
Each of these aspects of who I am and influences how I think about and interact with my research. Feminist pragmatist theory argues for the importance of experience in constructing knowledge (Seigfried, 2002) and the impossibility of separating theory from practice. Sandra Harding (1992) shares that only by acknowledging our positionality and relationship to the work we do can we begin to be objective. In an effort to situate the knowledge I'm constructing, and sharing my experience and relationship to it, I share my relationship to this work in my positionality, and throughout each chapter. My relationship to teaching, and the experiences I have had in the classroom as a student, and an educator have influenced how I pursued this work, my understanding of the work, and my interpretations and findings of this work.

I am a pragmatist, and I believe that my theory and my practice cannot be separated. If I see areas of the world that should be improved, I believe I learn about theory through doing it. In my practice this means that the work I do in computer science is in the hopes of building a better future for students new to computer science. A future where they are free to be who they are, and where they will build a future where we all belong.

In this dissertation I am breaking from the traditional academic convention, and including my own experiences and motivations in the chapters I present so that I could share and critically examine my positionality in relation to this work.

Chapter 1

INTRODUCTION: BUILDING A FUTURE FOR EVERYBODY



The times I have felt most joyfully triumphant in my academic and professional world are two. First, I felt like the *Queen of the World* when my code compiled in my first computer science class. I have felt similar euphoria other times whenever I have debugged a particularly difficult problem. Second, I feel like I am *Queen of the Classroom* when all of my students are learning and laughing and proud of what they are doing. The circumstances leading to this particular scenario are generally wildly chaotic, loud, messy and joyful.

Stepping back, that intro CS course was miserable. I felt lost the entire time. There were assumptions made of my previous knowledge. The professor deeply loved CS and would talk about how you were really only a computer scientist if you knew all the key commands and did not need a mouse. He would spend ten minutes lecturing how three planes could come together to make a point, then jump over the complicated concept of pointers without any explanation. He assumed that everyone knew or had many of the experiences that he had, assuming we inherently knew how to find the terminal on our machines and had spent time playing around with it. Even though I'd picked parts, saved for, and successfully assembled my own computer in high school. I did not have confidence in those experiences, so his examples and explanations did not make sense to me. I still struggled to find my terminal. I didn't understand his passing references to

brands and models of cars, or mowing the lawn. Those were not reference points I could build my knowledge on. I constantly felt behind and lost.

When trying to complete any work, I inherently felt dumb, and there was no way to tie what I was learning to previous knowledge. The intro level course assumed I had knowledge and experiences that I did not, but that many of my peers did. The only way I passed my first lab was by relying on a generous classmate who walked me step by step through missing explanations. I was deeply thankful for his patience and kindness, but it was a complicated power dynamic. He had asked me by email to study for a physics final the previous semester, sharing that we were in the same physics lab. I had no idea who this guy was, but I was down to study at the underground undergraduate library on a sunny afternoon. Shortly after we arrived, I realized this was more of a date than a study session. He peppered me with questions about what kinds of cars my family drove and my political affiliations. I didn't know how to respond, but politely answered his questions and kept redirecting us to the task at hand. I declined a second study date, but remained friendly. Because this classmate was in my major, which meant we were taking all of our courses together in sequence, he was in my computer science course. We lived in the same dorm, and when I was desperately trying to figure out my lab, he noticed and offered help. These interpersonal complexities followed me into adulthood as he continued to express romantic interest. He was so generous with his time, and I felt guilty constantly reiterating that I was not interested in dating.

Office hours were not helpful. Everyone assumed that the course would be miserable. It was miserable. We learned MATLAB, some other language lost to memory, and C++ in the course of 14 weeks. I constantly felt like I was behind because we had so much to learn so quickly. None of these experiences of this CS class match with my experiences of joyful classrooms.

The reason I felt like the *Queen of the World* in this class was that I had cleared a whole week to work on this CS final lab in the windowless, sterile, cold, smelly basement of the CS building. Every other assignment in this course felt impossible and arbitrary and I couldn't wrap my mind around how I was going to pass the hardest lab. I chose office hours to get started because I was going to need help. I sat down to work on the alphabetical grocery sorting algorithm, stopping

to questions: why was I sorting my grocery list alphabetically when you go to the grocery store, you sort your list by section. It felt like whoever wrote this assignment was not grocery shopping for their household, and the entire assignment felt arbitrary.

I wrote the program, and for anyone that's never programmed in C++, you have to make (compile) the actual file you want to run before you can check or run any of your code. It adds one step of confusion and required patience between you and debugging. This program was notoriously difficult to compile. But on my second attempt, the code compiled and my groceries were sorted. I felt elation and confusion. First, I was relieved the code had compiled, second I couldn't understand how I could have written correct code so quickly. I was so bewildered and surprised that I waited in line, doubting my self and my solution, to check with the office hours TA, because how could it have been so simple? I must have done something wrong. The TA was confused why I was asking and was like, "Of course it works," and blew me off to answer someone with a more important question. I walked out of that lab on air with an unexpectedly free afternoon before me and the world in my backpack, relieved that I would never have to program again, if I could avoid it.

As a teacher, I felt like *The Queen* when I would look around my classroom and each student was chasing their own curiosity, after working together to design the courses, and even then going beyond the requirements they set for themselves because they were so interested in what they were doing. It felt like the students were learning hard things with joy and enthusiasm, and my job was to coach them and help them think about ways to grapple with challenges they were encountering.

In one particular *Queen* moment, I was teaching middle school makerspace in the mornings and high school engineering in the afternoon. Students also had free time to use the makerspace at lunch, and could also drop by and work during class for class projects, or afterschool. The makerspace was an open space with an attached classroom with laser cutters, sewing machines, 3D printers, cardboard, and supplies scattered everywhere. Students had open lockers to store their projects. For some unknown reason at one point, there were 4 different projects focusing on hands. One was an engineering student who had rigged linkages modeled after a drawing from

da Vinci's journals as part of a history of engineering unit. One was a middle schooler who was making a hand to pick out books for English class. Another student was modeling a wearable infinity gauntlet for a physics course. He was wiring it with LEDs and sensors so that it would light up when he flexed. Another student was working on modeling prosthetic hands as part of an independent study he initiated when he found a match between a community need and resources that he had. Each of these students was working independently and they were working through frustrations and rising to the challenges they presented and defined for themselves. I was just there watching the community, the joy, and the whirring all around me.

Of course, I was also answering one million questions (as any caretaker does) but I wasn't driving the learning, I was observing it and supporting it. I also often did not know the answers to the questions I was being asked. I got to practice saying "I don't know" over and over again. I got to develop the addendum, "but let's figure it out together." And I have never learned in so many directions at once.

I got to learn a lot of JavaScript (p5js is a magical tool with great documentation and immediate feedback for teens.) I got to learn how to tear down and rebuild a 3d printer - because everyone else had to go to class and I felt the responsibility and the urgency to fix the machine. I learned how to build a chip with only logic gates because a student was curious about the inside of his mac. I never knew what questions I was going to get, and I never knew if I was going to be able to answer them, but my focus wasn't on knowing, my focus was on helping kids do things they cared about so the not knowing didn't feel like a failure, it felt like a step in figuring out what was next.

Students of course had different responses when bumping into unknowns. Some shut down. Some started with tears. Some turned to anger. Some didn't flinch. Part of my role was supporting those different emotional paths that students brought into the process. Each student had different experiences, different baggage, and different starting knowledge. They also had different interests, different goals, and different things they cared about. No two questions were the same, even if they were asked in the same words. I had to rely on relationships I had with those students, trust that I had in those students, patience, and trust in myself that we would

find or learn the skills we needed and that we'd figure it out together. It was noisy, chaotic, beautiful. There was never enough time in the day or enough of me to answer all the questions.

The longer I taught, learned about teaching, and by teaching, the more important and organic it felt to distribute that power I had. At first that meant taking small steps in a very traditional school, like exploring with project based learning instead of tests. In one geometry class, I realized we could cover the content in the tessellation chapter by creating friezes that included translations, rotations, and reflections. In the first year students sourced and used paper and blocks and legos and those sorts of things. In subsequent years, they got more and more creative. One student even wrote a piano piece because she realized that at a certain point on the musical register the notes flipped upside down, rotating 180 degrees. Another student made an entirely edible frieze out of rice krispie treats for us to share on the day the project was due. The projects became events that students saw older peers doing, and looked forward to doing themselves. In a physics class, a student Blake¹ heard that we would be making musical instruments in the spring, so on the first day of class in August, he asked if it 'would be alright if [he] built a synthesizer from scratch?', and 'would it be ok if [he] started early at home?'. Blake spent 8 months learning details of electrical engineering, programming, and machining parts in order to demonstrate his mastery of wavelength. He was so excited that I received regular updates about how this chip worked or this resistor was too small and got too hot to be safe, and this capacitor blew up. Fortunately for me, he had very supportive parents who were thrilled he was working on this. He had a fully functional synthesizer in April to show off when the project was due.

I found out during staff meetings that Blake hadn't submitted a single history assignment in weeks, and his teachers were struggling to get him to pay attention during class. I didn't have to fight those battles with him. He was motivated to learn new and complex topics and to make that synthesizer. He was engaged in class, and rarely forgot to do his problem sets, and was always deeply apologetic when he did. Blake was not a perfect student, but he was joyful in my classes and so fun to debate physics, astronomy, programming, and marvel heroes with.

¹I've used pseudonyms for all students.

Slowly, my classes shifted to be more and more oriented about projects and less and less oriented about tests. I moved from the state of Kentucky, where my teaching career began, to Boston where I worked in a school that encouraged project based learning. Computer Science was taught as part of sophomore geometry, and I taught many of the geometry classes. Sometimes students would request tests or exams alongside projects because they were worried they weren't learning enough with their projects. Many found they were surprised that the projects were more work than tests, but also more enjoyable.

Eventually, I developed a model of co-construction for my classes which I was able to practice for a few years in Boston and academically formalize (Everson et al., 2022, 2024). As a teacher, I would work to collaboratively create assignments and their requirements with my students. It allowed us to decide together how we would cover a topic. It also allowed for us to design around individual interests and needs. We were able to design projects and courses that were relevant to students, with choice, and a variety of entry points. We could collectively decide how to gauge what students were learning.

I really appreciate how my labmate, Eman Sherif, articulated co-construction:

Co-construction expands on constructionism (Papert & Harel, 1991), but utilizes collaboration between the teacher and student to create knowledge. In co-construction teachers and students engage in discussion, negotiate meaning, and arrive at a common understanding for an issue. Co-construction argues that knowledge is generated socially through interaction, scaffolding, negotiation, and feedback instead of knowledge only being passed from teacher to student. This collaborative process can involve peer interaction, group work, or designing curriculum together. By grounding learning in social interaction and shared agency, co-construction can foster deeper conceptual understanding, improved critical thinking, stronger communication, and a sense of ownership over learning.

She so clearly captured embodied pedagogical practices I've developed with students, and struggle to articulate, and that she's practiced in her work with students (Sherif, 2025).

There must be some way to create computing experiences between these two examples one where I can have the joy of that program perfectly compiling, but also one where I can have relevant and meaningful examples, and be driven by my own curiosity, and supported by my instructors, also having a joyful experience.

Computing in K-12 does not always take the same form that computing takes in post-secondary settings. It might be part of a language lesson, or appear in an art class, or underpin a Geometry class. It isn't uniform and does not always appear in a formal computing course. Secondary computing also does not have the capitalistic extractive history the formal computing carries. Instead it is often under resourced and underfunded. Teachers are often learning computer science as they teach it, learning as their students learn. It is also still being adopted in secondary schools, and isn't ensconced the way other subjects are. This offers the promise and potential that we can design for joy.

The world surrounding these educational experiences I'm describing is in the western hemisphere and the global north. It is white and western capitalistic with many judeo/christian traditions. It is inherently patriarchal. Education is usually underfunded. This false scarcity means that there is fierce competition for resources and seats at schools.

Because we live in a such a world, I believe that unless we are making efforts at every step to understand and create an intentional alternative culture in CS education spaces, we are perpetuating the exclusionary assumptions and problems that exist. I've also witnessed those who are successful in patriarchal classrooms continue to be the next generation of classroom creators, and they often recreate the classrooms which have made them successful.

We are all part of the future. And we need all perspectives and people building that future. Otherwise, we will continue to design for the current socio-cultural systems that privilege white, male, able-bodied, heterosexual, cis-gendered experience (Ko et al., 2023). Such design defaults risk excluding many people and many lived experiences. With only 20 percent of computing degrees going to women (Cheryan et al., 2023; for Science & (NCSES), 2023), it's clear that not everyone is welcome to build the future. We need to imagine new inclusive futures for secondary computer science education: one where all students are excited and engaged in constructing their

own knowledge and keep learning.

Students can learn through suffering, and joy is not necessary for learning, but joy is necessary to want to participate and to continue learning (Jane Margolis et al., 2017). It helps students to persevere and push through the struggle (Roque, 2024).

But beyond just more effective learning, students and teachers deserve a joyful learning experiences. In my experiences, learning requires hard work, and can be incredibly frustrating and infuriating, and it can also be very joyful.

There must be pedagogies that lead to joy in computing² where students are free to do what they want to do, and to learn how they need to learn. In order to test this vision, I've sought to understand joyful computing environments for teachers and students, to observe commonalities of those spaces to understand if there are common elements in these spaces by asking:

- RQ: What do joyful CS learning spaces have in common?
- RQ: What pedagogies support joyful CS learning spaces?

In this work, I will explore the thesis:

Feminist pedagogies encourage joyful secondary computing education.

The work described in this document seeks to better understand joyful computing spaces and the pedagogies that support them.

In chapter 2, I will describe my approach to this work as well as the foundational work and pedagogies that I build upon. I will also explain why I needed to do the following studies to begin to understand joyful computing.

In chapter 3, I will examine and deconstruct the concept of rigor through interviewing CS instructors, students, and secondary teachers. In doing this I will share some of the default assumptions and defaults the concept of rigor brings to broader perceptions of good CS education, and examine how they influence belonging in computing.

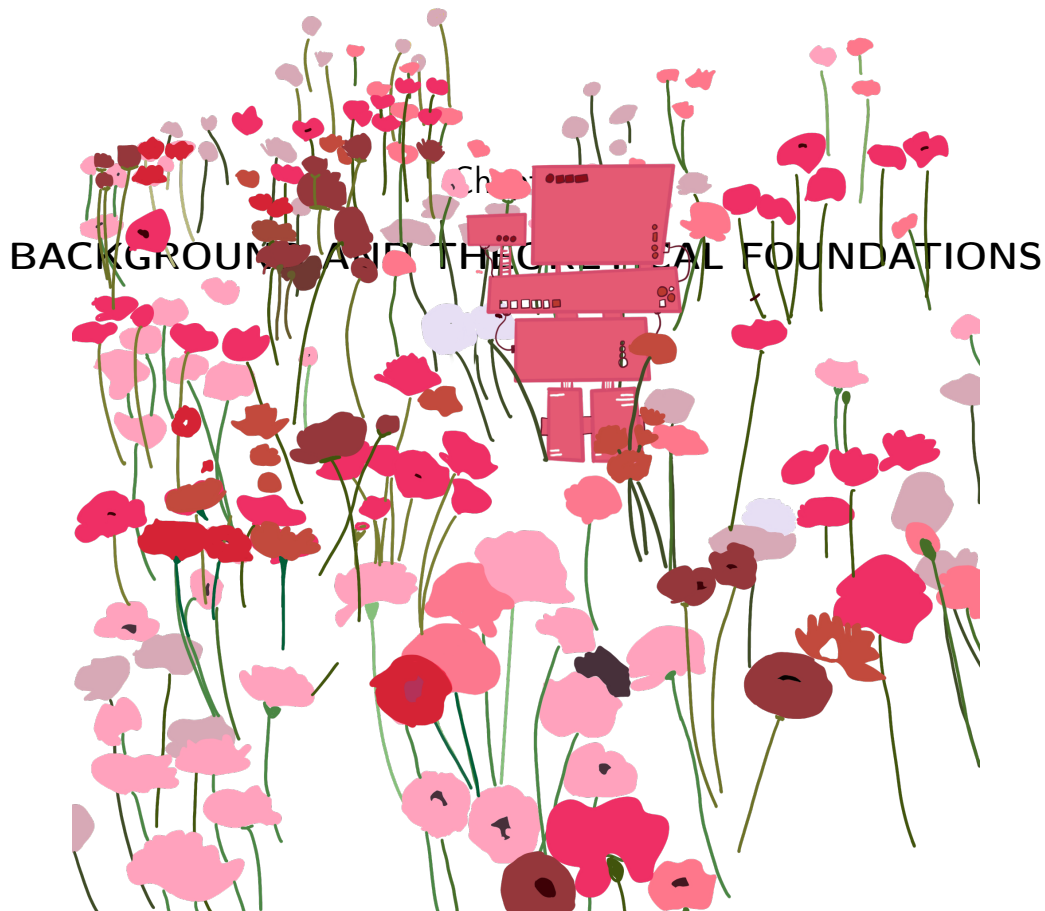
²Although some scholars articulate a difference between computer science, CS, and computing, in this work, I use these phrases interchangeably as I have seen it used as mechanism for gatekeeping, to invalidate some people's experiences of computing, and I believe it is important to let individuals articulate their own experiences.

Then in chapter 4, I will examine what students' perceptions of what CS classes which align with many of the masculine defaults surfaced in chapter 3. I will also examine what they imagine joyful CS to be, and what they want that experience to be like. This work was done with the University of Washington Youth Advisory Board. They were generous in sharing their thoughts and opinions, and also confirming our research findings. I will highlight the needs and the pedagogy that students think creates those ideal environments.

In chapters 5 and 6, I will share two studies of experiences of joyful computing, these studies also allow us an opportunity to re-imagine and revisit what rigor could be. First in chapter 5, we will examine joyful CS experiences from the student perspective. I interviewed students who identified their secondary learning environments as joyful. I also interviewed 5 teachers those students nominated. Here I surface the associated pedagogies that supported those joyful experiences. Then, in chapter 6 we will hear about teacher's perceptions of joyful computing and read about their experiences of joy, triangulating between speculative designs by teachers and through and design probes that teachers completed throughout their school year.

Next, in chapter 7, I will test the pedagogies from the previous chapters from an embodied position by examining computational embroidery and its associated pedagogies as an experience of joyful computing. In this chapter I share research with a summer course teaching computational embroidery and share students' work, perceptions of learning computing, and instructors' perceptions of teaching. As instructors, we designed this course to challenge some of the dominant assumptions of computing curriculum and students used the freedom of that course to learn computing, entangle their identities in their work, and make beautiful artefacts.

And finally in chapter 8 we will discuss how these studies build upon each other and support my thesis, that *feminist pedagogies encourage joyful secondary computing education* and how this work contributes (1) commonalities of joyful secondary CS spaces and (2) a recommendation of praxis and pedagogies for joyful CS education.



2.1 Grounded in Joyful Learning

I know joy is important in learning because I've watched it fuel my own learning and I've seen it fuel my students' learning. I watched students discover the joy of creating something they decided to make, problem solving the things they cared about, defining what they wanted to learn, and figuring out how to learn it.

My first education degree was mostly focused on classroom control and making sure I knew how to hold students in line. We had entire assignments where other pre-service teachers would pretend to be problematic students and we were graded on whether or not we responded with the appropriate level of authority and firmness so as not to lose control of our classrooms or the respect (really fear) of our students. This all happened while I was loosening the reins in my

own classroom, and showing students a different kind of respect.

Let me explain this by telling you about Evan. Evan had a lot of energy, was full of kindness, was brilliant and curious, and loved math. Once, when a teacher pretended to die with grief at a lack of student effort and interest, Evan, without a second thought, had transformed into a velociraptor, jumped over his desk and pretended to eat the teacher's corpse. It was obvious to me that Evan needed a little (or a lot) of flexibility with how he sat while working. I didn't care if he did his work standing at the counter in the classroom, or if he curled up in his chair. However, the school demanded that Evan sat with his "seat in the seat and feet on the floor." This is one tiny example of autonomy that I could offer Evan that offered respect and a tiny bit of choice to his daily learning experience. In broader freedom, I was able to offer Evan choice in how he pursued what he was curious about, and how he demonstrated his knowledge. In offering this respect, I often faced criticism from administration because my classrooms were loud, and "disruptive" because we danced when we learned the slope of the line, and my room was always a chaotic mess at the end of the day.

In my second education degree, I got to take a class called *Designing for Learning by Creating* where I read excerpts from *Mindstorms* by Seymour Papert and *Teaching to Transgress* by bell hooks (hooks, 1994; Papert, 1980). I finally had language and theory to describe what I was witnessing in my classroom. The way Papert described the teacher learning alongside the student, and the way the student realize the freedom they had when the teacher didn't know the answer either described some of the experiences I had working with students when they were building instruments I'd never seen before. The way hooks describes the excitement and joy that comes from respect and love in the classroom when she described *engaged pedagogy* described experiences I had co-constructing classes with students (hooks, 1994). These writings gave theory and validity to my lived experiences.

2.2 What is joyful learning?

In framing this work, I'm building on Ricarose Roque's conceptions of Joy as an Aesthetic (Roque, 2024) and a valuable end goal (Roque, 2023). Roque frames joy as something dynamic that can

coexist with struggle, trauma and grief, and is how humans are restored, sustained, and energized. It is a complex joy, *"Not joy as positivity or happiness, but joy from transformation, that emerges from the challenges and struggles they encounter as they create projects"* (paragraph 3). Her team intentionally physically designs for joy when working with families by configuring spaces to communicate belonging, and setting the environment with food and music. Her team also designs for intellectual joy by centering choice and agency, valuing funds of knowledge, and including familiar materials next to new technological tools. Facilitators work to de-center their own authority and strive to be co-learners and co-creators alongside children and families they are in community with. Similarly, Ahmed (2000) argues for the co-existence of conflicting feelings, and that joy can be a shared communal embodied feeling, but that it must be authentic. Montgomery and bergman position joy as a move from *"conditioned habits, reactions, and emotions"* and a growth in *"capacity to do and feel new things"* (Montgomery, 2017, pp. 59–60). They position thinking and feeling as one act in building new understandings and a new world.

I'm also building on Gholdy Muhammad's value of joy as an educational pursuit that exists in spite of system cultures of racism and oppression. She states that, *"children and teachers deserve a greater emphasis on joy through curriculum and instruction that is more invigorating and rigorous to cultivate the intellectual faculties of youth"* (Muhammad, 2022, p. 197). Additionally, she explicates how joy is often valued in primary schools, but is neglected as children age. This makes the pursuit of joy so much more valuable in secondary schools.

In this dissertation, I did not define joy for any study participants, given that it manifests in different ways for different individuals who have different lived experiences. Instead, I left it to each participant to define joy for themselves as joy is an embodied and emotional experience that can only be described by participants themselves. When describing joyful experiences, participants shared that they learned complex content and articulated the pedagogies their educators used to support that learning.

2.3 Miserable Learning (Hegemonic defaults in STEM and computing)

As a woman in engineering and computer science, I have constantly felt behind. Early on in my educational experiences, I assumed it was my fault that I didn't understand what was going on, or that I wasn't interested in the same hobbies as my classmates, like racing or hacking. As I got older I realized that there are norms and assumptions in STEM that prevent all students from engaging in joy.

In computing there are hegemonic and masculine defaults. These are cultural biases, norms aligned with hegemonic identities that prevent anyone who does not perform as this role dictates, from entering and sustaining in a field traditionally dominated by white and Asian men, like computing. Other defaults include self-reliance, competition, confidence, brilliance, risk-taking, and leadership (Cheryan & Markus, 2020). If someone fits the role of leader, and brilliant and confident dictated by these defaults, they are then associated with meritocracy, and individuals performing them are perceived as more worthy, regardless of ability.

Hegemonic defaults in computing, mean that many students will have early computing experiences like I did. Ones that are not collaborative or inviting, with sterile smelly windowless computer lab (2023). In CS culture, hegemonic defaults also present specifically as a love for precision, abstraction, aggression, machines, and a disdain for behaviors or ideas that may threaten the marriage between masculinity and technical competency (Carrigan, 2024). Early computer experts constructed masculine identities that valued individual genius, eccentricity, and isolation (Ensmenger, 2015).

These defaults influence each level of culture: ideas, institutions, interactions and individuals (Garr-Schultz et al., 2023), and play out as harassment, and lower perceived competence of women that also influence the diversity of computing (Patitsas et al., 2014).

These norms continue in secondary computing environments. Margolis (2002) points out that these defaults present in secondary computing as an "all boy club" where young men are free to play, and young women are often discouraged from engaging in computing because of cultural norms of play. In a study of middle schoolers' computing interests, Ashlock and Tufekci (2026)

found that young girls are less likely to enjoy traditional constructions of computing, and argue that it is important to change how the field itself communicates what it is to youth. Teens hold perceptions of that CS is "hard," for men, and "for people who are very smart" may influence how young people construct their identities and make plans to explore computing (Wang et al., 2016). Additionally, when researchers studied whether students perceived they "fit" in CS, girls were less likely to see themselves as skilled in the areas discussed above as computing defaults (like skilled in math or science) and saw themselves represented in media less than their peers (Wang et al., 2017).

All levels of culture are communicating that CS is for an exclusive group of people defined by these hegemonic defaults; broadly, in the media as a whole, institutionally in the requirements we set for programs and which students are admitted to those programs, at the interaction level in classrooms and schools, and at the individual level in how students' perceive themselves.

2.4 Joy in Pedagogy

As a teacher, I made thousands of decisions a day about my classroom: how to present what I was teaching, how to respond to a question, how to grade an assignment, where to stand, how to handle students who were uninterested in being in geometry. To make each decision, I had to balance students' needs, my abilities and knowledge, school policy, and the ethical implications of any decisions I made. Guiding those decisions are what I believe about the world (ontology) and how I believe we learn and construct knowledge (epistemology). Those decisions come together to form my pedagogy. Pedagogy in education is the practice of teaching, and the decisions a teacher makes in the classroom. It is the embodied practice of what a teacher believes about the world and how a teacher believes about how knowledge is constructed. Pedagogy influences the environment and the experience of the classroom.

There are a number of pedagogies that offer the possibility of joy and the freedoms that come with it. As I was thinking through this theory section I grappled with a number of them, and since I understand things by creating, I made a giant venn diagram in Figure 2.1 to begin to make sense of different theories and pedagogies.

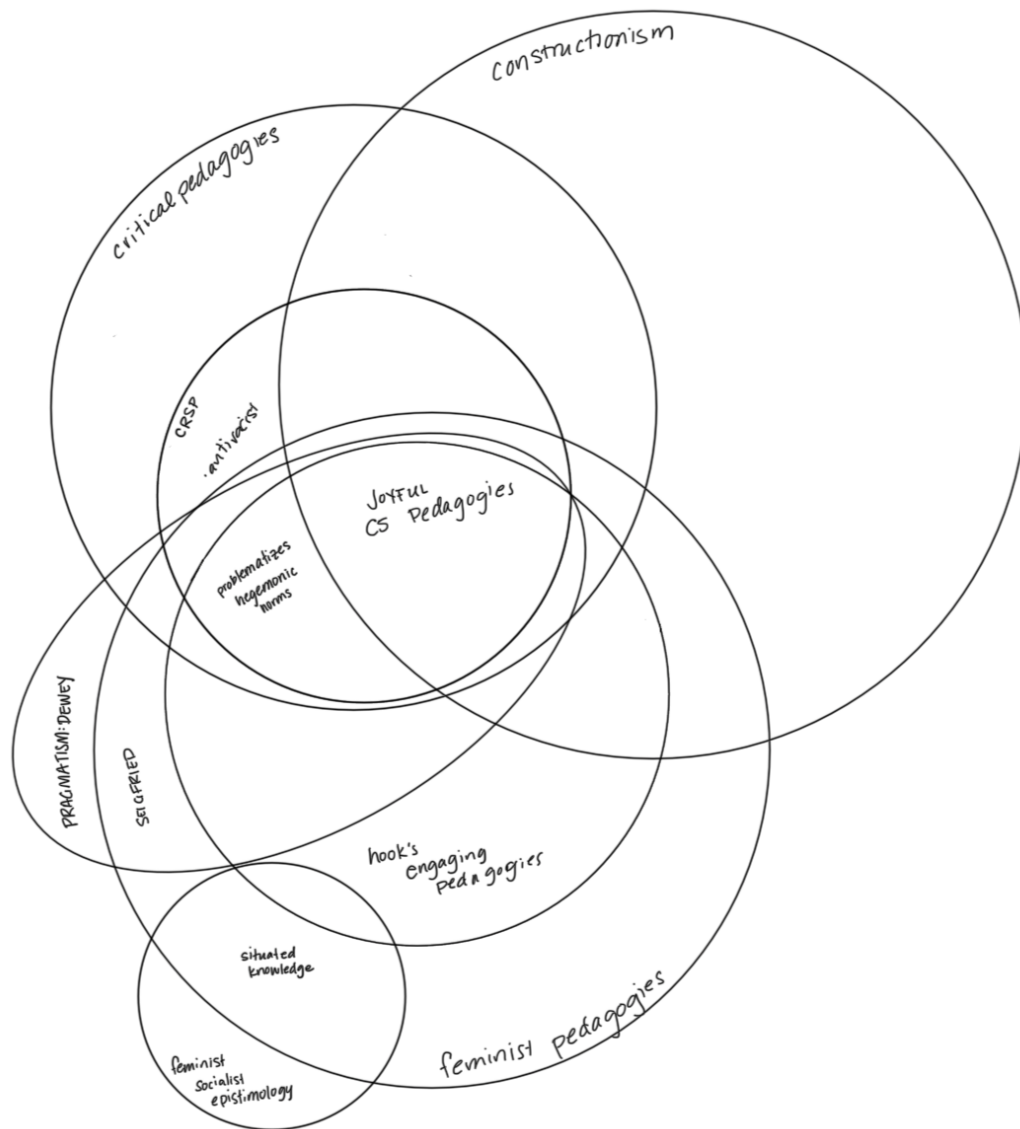


Figure 2.1: Theories and frameworks overlap and many theories carry the same pedagogies. The goal of this study is to identify which pedagogies encourage joyful learning.

2.4.1 Joy in Constructionism

Constructionist pedagogy encourages students to build their knowledge in part by creating artifacts of personal meaning. Constructionism suggests that students should bring their interests and passions to engage with artifacts. Applied to pedagogy, constructionism argues that a child's learning is amplified when creating something for others to experience (seen, heard, felt, etc.) (Papert, 1980). Constructionism theorizes that children construct knowledge through interactions and experiences with and in the world around them, and by building to better understand.

In its epistemology, Sherry Turkle notes that constructionism offers the opportunity to run counter to the norms of computer science and programming, and offers a "concrete and personal approach to materials" (Turkle & Papert, 1990, p. 1). Learning here happens by doing. Because learners are defining how and what they do, constructionism offers the potential to do work counter to cultural norms, which in computing, default to the masculine. In doing so, constructionism offers the potential for joyful learning.

Constructionism in computing education

In computing education, constructionist tools are often used to teach computer science in a way that lowers cognitive load for students (Federici & Stern, 2011). Some center materiality and physicality in computing (M. J. Johnson et al., 2023). For example, the Arduino (Luciano et al., 2019) and LEGO-robotics (Papert, 1980) are constructionist approaches that integrate physical making, creativity, and computing.

Prior work on constructionism and computing education has shown that constructionist pedagogy is a powerful way to teach computing while allowing students to express their interests and apply their knowledge of the real world (Tissenbaum et al., 2021). Researchers and practitioners found a variety of ways to do this. Horn et. al. (2020) designed *TunePad*, a platform that integrates music and computing. They found that the platform empowered a diverse set of users to engage with their interest in music while learning to code, developing their interest in becoming programmers. Additionally, *Quilt Snaps* developed by Buechley (2005) is a programmable

component that can be added to textiles. They found that presenting computing alongside students' existing interests motivated them to take more computer science courses. Buechley et. al. (2008) built on this work and created the *LilyPad* kit, which included a microcontroller and an assortment of sensors and actuators that increased engagement as students remixed the kit to create aesthetically custom wearable items. Keune has found that the materiality of the mediums that students have access to matters, and there are many understudied textiles arts to teach computing (Keune, 2022b, 2024). In other examples of constructionism, students learn to operate 'real tools' like telescopes (Azevedo, 2013; D. A. Fields, 2009). Ultimately, constructionism allows students to create personal and meaningful artifacts to do work that matters by sharing with their communities — whether it be family, classmates, online forums, etc. (Tissenbaum et al., 2021). These studies have done an excellent job incorporating student expertise, experience, and identity, and have not yet explored what it would mean if students were also to define their own assignments and assessments or were involved in setting their own learning goals throughout the course, exploring educational freedom and creating their own joy.

2.4.2 Joy in Critical pedagogies

In his work *Pedagogy of the Oppressed*, Paulo Freire argues that we dismantle the 'dehumanization' in world through education. The goal of education is so that students can become "beings for themselves" (Freire, 2017, p. 47) "To surmount the oppression, people must first critically recognize its causes, so that through transforming action they can create a new situation" (Freire, 2017, p. 21). This transformation comes through dialogue.

In critical pedagogy, trust is essential, and education replaces the "banking model" with dialogue. "Education must begin with the solution of the teacher-student contradiction, by reconciling the poles of the contradiction so that both are simultaneously teachers and students (Freire, 2017, p. 45)."

This shifts the direction of the content in the classroom to that where all members of the classroom contribute. Critical pedagogies also value students' previous lived experiences that should be considered assets (Moll et al., 1992).

Critical pedagogy's ontological vocation is humanization and that people are "beings in the process of becoming" (Freire, 2017, p. 57). That promise of becoming offers freedom that has the potential to incorporate joy into learning.

Additional critical pedagogies that challenge hegemonic norms include decolonial pedagogies that imagine a world without colonialism and seek to center and value indigenous knowledge (Karetai et al., 2023), and indigenous pedagogies (Kimmerer, 2020; Leddy, 2024), which highlight alternate ways of knowing such as the importance of respect, relevance, reciprocity, and responsibility.

Critical pedagogies in computing education

There are several specific pedagogies which build on critical pedagogy in computing education. Many frameworks aim to incorporate students' lived experiences into pedagogy. Culturally Responsive Pedagogy (CRP) is one way researchers and practitioners have tried to address the access gap in computing education (Margolis et al., 2012). CRP relies on students' experiences and cultural knowledge (Gay, 2018). It enacts instructional methods that prioritize student-teacher relationships, classroom climate, and materials that support student learning. CRP is a teaching approach emphasizing academic success, cultural competence, and critical consciousness (Ladson-Billings, 1995). It is designed to empower students by connecting learning to their cultural contexts and, therefore, their lived experiences. She has also examined the work since then, articulating the importance of finding the possibilities that each student holds when allowed to bring their full self into the classroom (Ladson-Billings, 2021). Culturally responsive pedagogy challenges traditional deficit models, which often dismiss students' cultural backgrounds and instead view these cultural elements as strengths (Freire, 2017). The approach encourages teachers to build meaningful connections with students, incorporate students' experiences into lessons, and foster critical thinking that challenges the status quo. CRP focuses on validating and sustaining students' cultural identities while promoting social justice (Gay, 2018).

Culturally responsive computing (CRC) builds on the principles of culturally responsive pedagogy by applying them to computer science and technology education (Scott et al., 2015). CRC

includes culturally situated design tools that connect computing concepts with cultural practices, such as African American hair braiding (Eglash et al., 2013) or Native American quilting (Y. Kafai, Searle, et al., 2014; K. A. Searle & Kafai, 2015), offering students an opportunity to explore both their heritage and computing. CRC seeks to promote digital equity, innovation, and social empowerment by integrating culture and computing.

Another framework is Culturally Responsive-Sustaining Computer Science Pedagogy (J. Davis et al., 2019) which builds on CRP and Culturally Relevant Pedagogy to ensure that "student's interests, identities, and cultures are embraced and affirmed."

Prior work has integrated culturally responsive pedagogy into computing education in many ways (Ashcraft et al., 2017; Charleston & Charleston, 2014; Cooke et al., 2019; Y. Kafai, Searle, et al., 2014; Nakajima & Goode, 2019; Scott et al., 2015). For example, Lachney et al. (2021) found that CRC requires uplifting multiple ways of knowing, going beyond engaging traditional educators to integrating knowledge held by community members. Additionally, the afterschool program COMPUgirls leveraged CRP by creating computing learning experiences that consider race, gender, language, and culture (Scott et al., 2015). I also taught a high school computer science summer course framed by CRP (Everson et al., 2022). The course presented students with counternarratives about access to computing education, algorithmic justice, and data privacy. Our research team found that students transitioned from counternarrative consumers to changemakers concerned about the perils of computing and motivated to mitigate them. These examples demonstrate that CRP is successful when it centers on community, addresses social change, and provides students with meaningful, context-rich opportunities to critically engage with computing. By fostering authentic relationships and situating students as agents of change, CRP not only makes computing education more inclusive but also equips learners to challenge systemic inequities and contribute to their communities.

Critical pedagogies in computing education do very important things like value the learner, and the learners lived experience, and work to deconstruct hegemonic norms in the classroom. These pedagogies also do not consider the embodied nature of knowledge construction, or the importance of autonomy in educational environments.

2.4.3 *Joy in Feminist Pedagogies*

There are several threads of feminist pedagogies discussed in this section; feminism, feminist pragmatism, bell hooks and Black feminist epistemologies, all of which offer the possibility of joyful learning. There are other feminist epistemologies and pedagogies, like feminist materialism or posthuman feminism, not covered in this work because many of the pedagogies overlap, and the epistemological nuances would take pages to discuss.

Like other critical lenses, feminism problematizes systemic issues and points out exclusions and oppression. A refusal to go along with hegemonic norms, like masculine defaults, can be a killjoy for some (Ahmed, 2023). The perception of feminists is often that they are full of rage (Slaton & Riley, 2024) and want to burn the world down. This seems ironic when the point of this work, which is rooted in feminism, is to understand and build more joy in the world. One problem with hegemonic norms, is that joy is not accessible to everyone, and that many are excluded from even participating, let alone joyfully participating. In order to build that joyful future, we must disassemble exclusionary cultural defaults.

In addition to acknowledging systemic problems, feminism seeks to value the individual, build community, and value lived embodied experiences. "Feminists argue for a just and civil democratic society that dissolves binary thinking and focuses on relationality, rights and responsibility" (Blackmore, 2019). Feminist pedagogy also values that experiences are formed by identity, and personal experience is central to learning (Parry, 1996). Feminism recognizes of the importance of personal experience as a source of knowledge and emphasizes the value of perspectives of people of different races, classes, and cultures in community dialogue are essential to pushing back on those systemic problems (Weiler, 1991). Webb, Allen and Walker add that the individual voice is a way of knowing and that it is important to value diversity of voices in order to challenge hegemonic views (Webb et al., 2002).

Learning through experiences is embodied and embodied experiences are messy. Compelling work suggests that joy should be a meaningful part of learning, and therefore a pedagogical goal, and that feminist pedagogies support such joy. Approaching this work from the feminist mate-

rialist perspective, Tamboukou argues for the importance of embodiment and affect in learning environments and that a "creative and radical education" allows learners to re-imagine themselves and the world (Tamboukou, 2018). This embodied experience also includes an individual's affect and experience (Ahmed, 2000), and emotions play an important role in learning (Vea & Jaber, 2025). Rosenblum and Harris (2019) taught literature courses that required deep community dialogue and engaged the "messiness" of embodied digital reconstruction. Whereas Woglom and Jones centered feminist materialist practices and encouraged learners to push against "educational rigidity and institutional repetition" in learning music in a space that students and instructors "co-produced." They found that spatial flexibility, and creation were essential in creating this joyful cacophonous environment to learn to make music (Woglom & Jones, 2024). The tools and materials we use in an educational experience inform that embodied experience (Peppler & Thompson, 2024; Peppler et al., 2025).

Learners claim autonomy in classrooms that value the individual experiences. Privileging the individual voice of students means that individual students are trusted to make decisions about their experiences (Webb et al., 2002). If theory and practice are entangled (Seigfried, 2002), the theory of privileging a student's voice also means respecting that voice and allowing students autonomy and choice in their learning.

Placing an importance on relationships and community in a feminist educational environment allows students to respect each other and participate as subjects rather than objects in the classroom (Parry, 1996; Shrewsbury, 1993; Webb et al., 2002). Educators become co-conspirators and problem-solvers alongside students (Rosenblum & Harris, 2019). This shift recenters the learner and the learner's experience in the classroom.

Part of building community is shifting the relationship between students and teachers to one where teachers are "decentered" and instead, cooperation for the collective are centered. (Parry, 1996). Like hooks, Weiler argues that feminist pedagogy allows for a more concrete praxis to Freire's by questioning of the role and authority of the teacher (hooks, 1994; Weiler, 1991). Woglom and Jones (2024) realized the need to limit adult intervention and shift from teachers to support staff in building joyful music learning.

Feminist pedagogy has a rich past on which to build. It has a strong and ever-expanding theoretical base, one that continues to add to its number and its impact (Weaver-Hightower, 2020). Feminist pedagogies offer an opportunity to create joyful computing environments.

Pragmatism and feminism

Pragmatism and feminism have many overlaps in approach. John Dewey is considered a father of pragmatism, which foregrounds the situated nature of knowledge construction. Dewey argued that knowledge is not a cognitive relationship between subject and object. The subject and the object cannot be separated, and our individual past experiences, which vary, affect our interpretations. We must approach things critically as past interpretations influence everything because our intellect is the sum of our lived experiences (Seigfried, 2002). Feminist pragmatism argues that our embodied experiences are how we learn and construct our knowledge, and must be considered.

However, before Dewey published *Experience and Education* (1938) and *Art as Experience* (1934), his graduate student Ella Flag Young published her dissertation, *Isolation in Schools*. In this dissertation, she highlights the importance of teachers, stating "as is the teacher, so is the school" (Flagg Young, 1900, p. 19), and arguing for the importance of relying on teachers' experiences and knowledge and providing freedom and autonomy for teachers—so that the students in school also have freedom and autonomy. Teachers must experience joy as well.

Feminist pragmatism also argues that the purpose of schools is to create democratic citizens and that schools should provide content that liberates the individual and frees the identity from the confines of educational rigidity (Seigfried, 2002). This rigidity can prevent freedom and joy from being part of learning.

hooks engaged pedagogy

bell hooks built on Freire's critical pedagogy and developed her own engaged pedagogy, arguing the purpose of education as the "practice of freedom" (hooks, 1994, p. 207). In *Teaching to Transgress* hooks emphasizes the importance of pleasure and excitement (synonyms for joy) in

education, pointing out that many traditional and critical educators, including Freire forgot the importance of the the experience of the education (hooks, 1994). Because experience matters, students must be seen as individuals and that instructors must value each individuals presence and contribution.

Like Freire, hooks argued that teachers must be learning alongside students. "Engaged pedagogy does not seek to simply empower students. Any classroom that employs a holistic model of leaning will also be a place where teachers grow, and are empowered by the process (hooks, 1994, p. 21)." To do that, teachers must be vulnerable and take risks that they would ask the students to take. Teachers must also practice personal well being because if they are unhappy they cannot support others to be self-actualized and learning (hooks, 1994)

hooks also argues that theory alone is not enough– and that practice is part of understanding theory, "no gap exists between theory and practice" (hooks, 1994, p. 61). She also prioritized incorporating students' and teachers' personal stories into the classroom because "experience links discussions of facts or more abstract constructs to reality" (hooks, 1994, p. 86). Each student becomes interested in participating when it relates particularly to them. Everyone benefits when we value the lived embodied experiences of students. Students' experiences are must all be valued and seen as assets.

This pedagogical work must be done in community and collectively. Community must be built in the classroom, and we need everyone for collective action to happen. This does not mean we cannot disagree, rather, differences should act as a catalyst for learning and thinking about things in a new light. Changing ones mind must be seen as "valid and legitimate" (hooks, 1994, p. 131). And that truth is discovered when the community has consensus.

Black feminist thought and epistemologies in computing education

In addition to building on bell hooks, Black feminist thought builds on the work of Patricia Hill Collins which positions "U.S. Black women as knowledge agents or experts of their own experiences," and that Black women are not inferior to white men, white women, or Black men (Rankin et al., 2024). It values the lived experiences of Black women, and recognizes differences

in experience. Six core constructs of this intersectional thought are relationality, power structures and how they interact, social inequality is not inevitable, social context of knowledge production, complexity is inevitable, and social justice (Rankin et al., 2024).

In CS education, Black Feminist Thought recognizes the need to design computing education and environments so that Black women thrive; proposing that if we are designing for Black women, all students will benefit. Pearson et al. have examined the role of Black women educators and found that Black women deeply care for their students and have high standards which seek to empower their students (Pearson & Leggett-Robinson, 2024). Other studies using these intersectional framings have examined the experiences of Black women in educational environments, and shared the sites of violence that Black women have experienced, finding that K-12 settings, primarily white institutions, and internships are racist problematic places and that “CS education is not a colorblind meritocracy. Rather, it is a “matrix of intersecting oppressions” (Rankin et al., 2021a), and that Black women may be leaving computing fields as a form of resistance (Rankin et al., 2021b). In informal settings, studies have sought to encourage Black and Latina girls successfully through online summer camps (Braswell et al., 2021) or by sharing how transformative justice framework has been successful in designing online computational groups (Erete et al., 2021). There are few if any published examples of Black Feminist Pedagogies or Epistemologies implemented or celebrated in the secondary CS classroom.

Feminist pedagogy in engineering

There is prior work proposing and examining feminist pedagogies in post-secondary engineering. For example, a cluster of four papers explain feminism and propose ways of incorporating feminist pedagogy into post-secondary engineering as a way to improve learning. They explicate six common beliefs of feminism; social justice, democracy, individuality of expression, responsibility, inclusionary thinking, and freedom and employ their own research methodologies to explore their experiences as women in engineering (Eschenbach et al., 2005; Lord et al., 2005; Waller, 2005a, 2005b).

Where other prior work examines a shift to feminist pedagogy in the engineering classroom.

Donna Riley (2019) has developed a liberative pedagogy for engineering after reading *Teaching to Transgress* by bell hooks and meeting with bell hooks to discuss the work she was doing in transforming engineering pedagogy. She worked to incorporate critique of concepts and texts into her thermodynamics classes and to bring a feminist perspective to her pedagogy. She often received pushback that her engineering classes were not technical enough—because instead of exclusively doing a large number of problem sets, students would also reflect on the impact of the projects they were analyzing. Other work examined ways women have incorporated feminist practices into their engineering teaching and work by resisting cultural norms and carving new paths (Ambrose & Hornig, 1997). Other work examines the mess in creating experiences in feminist teaching (Parvin et al., 2022), and introducing playful feminist toys into dialogue and discussion (Parvin & Rouse, 2024) recognizing and making the lived experience of dialogue, discussion and learning more joyful.

Feminist pedagogy in computing education

There's a surprisingly limited amount of research using a feminist pedagogies in CS education research. Often it is mentioned in passing, or selected as a keyword in an article without any explanation or examination of feminist theory or pedagogy. The words feminism or feminist is often briefly mentioned and not further used or explored. Feminism is mentioned as part of a course reading (Bettin, 2023; Mayhew & Patitsas, 2023; Smith, 2024) or as a framing studied as part of a course's content (J. Davis et al., 2019; Gautam et al., 2024; Kumar & Kremer-Herman, 2019; Petelka et al., 2022). Feminism is also occasionally mentioned when researchers are exploring the experiences of women in computing. Sometimes it is mentioned as a recommendation for future work (Mishkin, 2019; Yates & Plagnol, 2022) to provide historical context (Frieze & Quesenberry, 2013; Patitsas et al., 2014), as an alternative equivalent to the word gender (Berglund et al., 2006; Greening, 1999), or as a qualitative researcher's field of expertise (Lyon & Clayton, 2021). Feminism is also mentioned in literature reviews or meta-analyses as an emerging theory (Delcourt et al., 2024; Kavas et al., 2020), undervalued methodology (Ross, 2023), or identifying confused problematic framings (Brown et al., 2023). None of these mentions of feminism use the concept

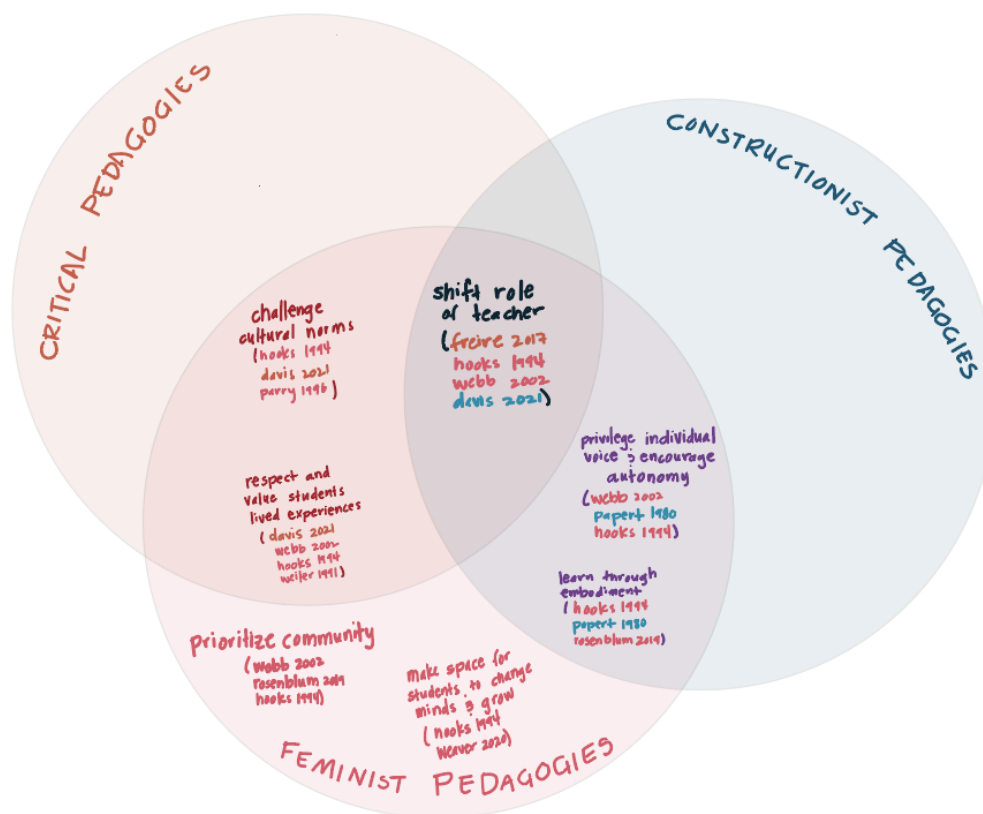


Figure 2.2: Joyful CS pedagogies are feminist pedagogies sit at the center of critical, constructionist, and feminist pedagogies. Feminism centers relationality as well as the embodied construction of knowledge, while acknowledging the systemic inequities of the world.

as a research framing or as an intentional course pedagogy.

When feminism is used as a research framing, it is often to used as a deficit lens. For example, Convertino (2020) used feminist post-structural theories (which critique gender as a pre-given identity) to study Latinx women's experiences in computing, and found that some experienced social isolation and also were frustrated by the perception of computing as a space of visible invisibility. Another study of educators found that biases and structures prevented women from entering engineering as a career, and that the profession suffers when it doesn't have diverse

voices (Beddoes, 2011).

When feminism is used as a pedagogical framing in computing, it is invoked to support marginalized groups to have a better computing experience. Ren proposes a feminist pedagogy that builds a multicultural classroom, emphasizes collaborative learning, and interdisciplinary work (Ren, 2022). Ocak et al. (2024) propose a theoretical framework employing principles from Feminist Standpoint Theory that includes dialogic and dialectic relationships, embracing intersectionality, and framing CS as a lived experience. This research offers theoretical ideas, but has not been practiced in a computing classroom.

In post-secondary courses, researchers found feminist pedagogies in courses and in informal learning settings for adults were found to be effective (Baynes & Norris, 2021), and to improve self-efficacy and initiative (Peña et al., 2021). Instructors have also used feminist pedagogies in computing ethics courses (McDonald et al., 2022; Trim & Gulley, 2023). However, when discussing ethics, researchers noted how difficult it was to find joy because of the nature of thinking about so many oppressive structures. They stated, “These are not happy subjects. So, for us, and for our students, joy comes from seeing subjects and identities and concepts given center stage that are not normally taught in computer science” (Trim & Gulley, 2023, p. 208). Ethics courses are often not part of the required curriculum, and are taught late in a program, leaving many students without the opportunity to study computing with feminist pedagogies.

I have not yet found studies explicitly examining feminist pedagogies in secondary computing classrooms. This could be because researchers and practitioners feel (as I also have felt) that they do not need to invoke the word feminism to live it. But this work seeks to fill that gap by examining joyful feminist pedagogies in the secondary computing classroom.

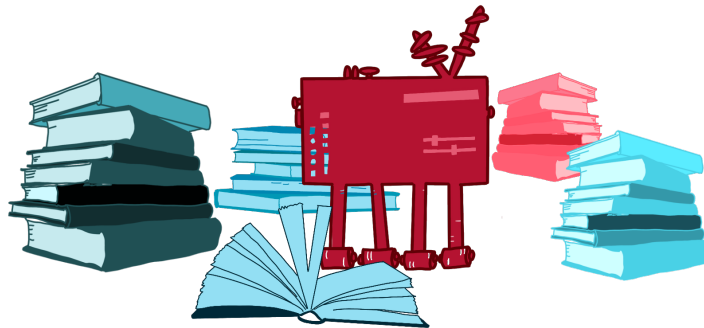
2.5 What could joyful CS pedagogy be?

Rather than approaching joyful CS education from what is missing, I'd like to approach it from the perspective of what could be. Of the pedagogies explored, constructionism offers us the opportunity to learn computing by creating things and sharing those things with community. Critical pedagogies offer us an opportunity to understand the injustices in the world, while valuing

the lived experiences of students. In addition to offering the embodied nature of constructionist learning, and the need to acknowledge the need for a justice centered pedagogy that critical pedagogy offers, feminism encourages us to imagine a better alternative world, and to learn about it by building such a future. See Figure 8.1. If we are to build such a future it is important to re-imagine it at each level, because as it is, each level currently prioritizes a particular way of being (Garr-Schultz et al., 2023; Lewis et al., 2016), rather than allowing all individuals to be free to be themselves (hooks, 1994).

Chapter 3

THE STATE OF CS: DECONSTRUCTING HEGEMONIC DEFAULTS IN COMPUTING BY INTERROGATING THE CONCEPT OF RIGOR



This work was written with analysis support from F. Megumi Kivuva, Eman Sherif, Al Oleson and Amy J. Ko. Work is always done in community. A version of this work was published in the Transactions on Computing Education Journal (Everson, Kivuva, Sherif, et al., 2026).

3.1 Importance of Understanding Perceptions of Rigor

I have a complicated relationship with the word rigor. It was something I always strove to meet in my engineering practices, but it felt uncomfortable, like a shirt that didn't quite fit. It required numbers, perfection, late-nights, sacrifice of all else and precision, while forgetting that humans were involved in the process. To do things with rigor I felt like I had to turn into a robot, and ignore my own experiences and curiosities. But it also felt like this level of expertise to aspire to, and a way to know I'd been doing things 'the right way'.

When I started teaching, I grew to hate the word rigor. I often heard it in staff meetings as a justification against more liberatory pedagogy. Teachers would argue that any changes to

the way we've taught math or CS would destroy the field and lead to the downfall of humanity. "How would we know that students understood the material if we weren't making them miserable with never-ending problem sets and tests where it was impossible to get an A?" Obviously I'm paraphrasing that last sentence, but I've found rigor is often invoked out of fear. Fear that we aren't able to teach our students well. Fear that we'll lose something if we pass power to the next generation. Fear that how we learned wasn't the best way to learn, and that we as teachers might not be the best at what we do.

When rigor was invoked, I saw more fear of a threat to identity hiding behind teaching content well. I wanted to understand what conceptions of rigor were, and if we could truly teach rigorous computing.

In computing, I saw *rigor* popping up in many spaces, and called as a necessity for quality instruction that improves students' experiences.. For example, Computer Science Teachers Association (CSTA), the organization whose standards are most widely adopted as state standards in the United States, works to "increase the availability of *rigorous* computer science for all students, especially those who are members of underrepresented groups" (Association, 2017). The National Science Foundation, which made possible much of the computing education research in the United States, "funds research and development that is building the necessary foundations for implementing *rigorous* and engaging computer science in schools across the U.S." (emphasis added) ("Computer Science Pedagogical Content Knowledge: Characterizing Teacher Performance: ACM Transactions on Computing Education: Vol 19, No 3", n.d.). Additionally, the Kapor Center Culturally Responsive-Sustaining Framework, which seeks to create equitable CS for all students, calls for *rigorous* pedagogy and curriculum (K. Davis et al., 2021).

All of these calls mention rigor, yet none of them defined what rigor is or what it should look like in CS education. There is no mission statement, no definition, no field-wide resource that defines the term "rigor." Yet, teachers are encouraged to teach rigorously and researchers are seeking to understand rigorous education.

However, if we, the CS education community do not understand what rigor is in instruction, or if definitions of rigor vary, at best, courses run the risk of communicating confusing practices

and expectations to students. And in the worst case, the community risks using rigor as a tool for gatekeeping and pushes students out of the field. It is unclear what rigor looks like in effective CS education, or how concepts of rigor play into equitable instruction. It is also unclear what students must do to achieve a rigorous understanding of CS and demonstrate they have achieved mastery of the subject. It is also unclear if what students think they must do matches what instructors think students need to do.

With such importance placed on “rigor” in standards, funding, and research, without clear agreement on what the word rigor means in CS education, and so many varied definitions of rigor in different fields, it is important to understand:

- What do CS instructors think rigor is?
- What do CS students think rigor is?

To answer these questions, I conducted 29 semi-structured interviews with 10 secondary teachers, 9 post-secondary instructors, and 10 post-secondary CS students.

3.2 Existing Conceptions of Rigor

I researched to understand how the idea of rigor has been used and understood within various disciplines, as well as how it impacts students’ experiences. This section starts with the liberal arts and move to the STEM fields and ends with definitions and uses of rigor in CS.

3.2.1 Conceptions of Rigor in Liberal Arts

To inform how CS education understands rigor, we can investigate how other fields have conceptualized rigor, starting with the liberal arts.

Researchers have found no consensus on how faculty define rigor. One faculty task force identified the “nebulous concept of rigor” as critical thinking, challenge, mastering complex material, time and labor, and credible work in university courses (J. E. Johnson et al., 2018). A follow up study found that faculty thought that upper-level courses had more rigor than lower-level courses and that students gave faculty higher rankings if their courses were perceived to be rigorous (J. E. Johnson et al., 2019). Additionally, when Draeger et al. conducted surveys and

focus groups with liberal arts professors to determine what professors thought was academically rigorous, they found four dimensions: active learning, meaningful content, higher-order thinking, and appropriate expectations (Draeger et al., 2013). Whereas one study of elite post-secondary institutions found instructional practices like active learning, high expectations, and tying learning to prior knowledge played a role in supporting rigor (Campbell & Dortch, 2018). In the liberal arts, professors held several definitions of rigor.

Students' understandings of rigor are also unclear. When examining student conceptions, Draeger et al. found that they differed from those of the faculty. Students perceived tough grading, workload, difficulty, interest, and interactions with instructors and peers as rigor (Draeger et al., 2015). A longitudinal study of college students operationalized two concepts of rigor—workload and instructor expectations. Workload improved student attitudes about learning, and instructor expectations improved students' self-motivated learning. (Culver et al., 2021). Whereas, another study found that students have figured out how to play the game of 'college management' not through hard work, but through choosing schedules, limiting workload, and interacting with professors: students were able to appear rigorous (Arum & Roksa, 2011). Students also held many conceptions of rigor.

Furthermore, some prior work argues that the concept of rigor harms students because the concept is used as a mechanism for gatekeeping. Nelson argues that traditional methods of rigor, like firm deadlines, content, and fairness, were "dysfunctional illusions of rigor" that prevented him from teaching well (C. E. Nelson, 2010). Jack and Sathy argued in *The Chronicle of Higher Education* that instructors use rigor as a way to "weed out" students rather than engage in inclusive teaching practices (Jack & Sathy, 2021), and Gannon notes in another *Chronicle* article that instructors use rigor as "cloaks to hide practices that actually erect barriers to student success" (Gannon, 2023). Cate Denial employs kindness as a pedagogical approach, believing the best in students, and seeking to listen to and meet student needs. Denial argues that education often sets students up as antagonists, seeking to avoid rigorous work, rather than making work relevant and interesting (Stommel et al., 2020).

In summary, prior work in the liberal arts conceptualizes rigor as: critical thinking, challenge,

mastering complex material, time and labor, credible work, active learning, meaningful content, higher-order thinking, appropriate expectations, tough grading, workload, difficulty, interest, interactions with instructors and peers, cognitive challenge, workload, firm deadlines, content, fairness, weeding-out, and gatekeeping. There are many competing conceptions of rigor, and in the liberal arts, it is unclear what rigor is.

3.2.2 Conceptions of Rigor in Science, Technology, Engineering, and Math

In STEM fields, there are additional definitions of rigor and those vary between disciplines. It is important to look to the domain-specific definitions of rigor by examining its use in education contexts to interrogate whether it reinforces hegemonic norms. In an examination of feminist perspectives in engineering education research, Riley argues that rigor is discipline-specific and that it is often a way to assert white male heterosexual privilege. She states "'Rigor' suggests a sense of being physically and/or mentally taxing, demanding." She then goes on to question who the field is making engineering hard for and points out that rigor often reinforces "gender, race, and class hierarchies in engineering," while also making invisible queer, disabled, low-income students (Riley, 2017). Therefore, it is important to understand how rigor is used in the domain-specific implementations of CS education, to ensure that CS education is not excluding students by relying on rigor.

In math, where I have prior experience teaching, there are even more definitions of rigor. One study found that math teachers often perceive rigor as higher grade level content in lower grade level classes and difficulty of problem type (J. D. Davis et al., 2019). My experiences align with this study. I watched teachers push more and more content into the year, often leaving many students behind. Instead of difficulty, other researchers attempted to define the tension between play and rigor because in the context of math, rigorous content is often characterized as all work, with little room for play (Wohlwend & Peppler, 2015). When I introduced play into my math classes, I got so much push-back and anxiety from other teachers and parents. They were worried my students would not be prepared for the next level of courses, or that their classes would be

perceived as boring (often appropriately).¹

Science education wrestles with the relationship between rigor and teacher expectations. A longitudinal study of the implementation of rigorous standards found that standards alone were insufficient to ensure student learning, and that teacher expectations about what students could achieve were also important (Bair & Bair, 2013). Additionally, a study of university biology students found that as students progressed in their coursework, they found cognitive complexity increasingly important. However, they also found that complexity was enabled by pedagogical practices such as alignment between assessment and instruction, active class practices, and faculty support (Wyse & Soneral, 2018).

In summary, from STEM we have additional conceptions of rigor: physically or mentally taxing, demanding, higher grade level content at lower grade levels, difficulty of problem, all work and no play, standards paired with teacher expectations, cognitive complexity, active class practices and faculty support. Again no clear definition of rigor arises from STEM.

3.2.3 *Conceptions of Rigor in CS*

In post-secondary CS education, rigor was often used as justification for curricular and pedagogical decision-making in the literature but was rarely defined. Sometimes it was mentioned in the context of mathematical correctness. While using mathematical games to teach rigor in CS, Ginat leaned on Webster's definition as "strictness". This study discussed rigor as a necessary problem solving notion, and used it to describe how game players examined the invariance of their algorithms (Ginat, 2007a). Other times, rigor had an essential role in making students into

¹Fortunately (or unfortunately), I had annual testing scores that showed my students were actually learning more compared to previous years. The first school I worked in was a private school that conducted annual standardized testing, and each year I checked the average geometry scores (where I was the only teacher) and our scores continually rose as I introduced more and more feminist pedagogy into my courses. I share this with the caveat that this is all from my memory, and this testing checking was driven by my own anxiety. I worried that I was under-serving students—because other teachers and administration regularly questioned my practices because they were not 'rigorous'. I sought every means to verify that I was supporting rather than harming students. In addition to checking test scores, I also solicited multiple teacher educators from a local university to observe and critique my practices. All feedback and data came back that students were well supported and learning.

better problem solvers and programmers, who consider programs broadly while debugging (Ginat, 2007b). One study set mathematical tools as one of three goals in programming, "fun, rigor, and pragmatism" (Sánchez-Calle & Velázquez-Iturbide, 1991). Course designers used the concept of rigor to legitimize and stratify post-secondary coursework. When working to build introductory CS courses, instructors developed a "rigor-first" approach, arguing that students must build a deep understanding of the "proper fundamentals of the discipline, including concepts, vocabulary, reasoning tools, and laboratory experience" (Dougherty & Wonnacott, 2005). Others argued that making CS learning both fun and rigorous could help retain CS majors (Dorodchi & Dehbozorgi, 2017). Similarly, Ramnath and Hoover presented a framework for balancing rigor and relevance in which they define rigor in two ways: first, as "the manner in which the methods of the discipline are adhered to" and second, "to denote the formal structures required by the discipline." They proposed to assess student learning to see if a course was rigorous, and survey students to see if the course was relevant, defining rigor as learning (Ramnath & Hoover, 2016).

In K-12 CS education, rigor was also used as an undefined aspirational goal. Sometimes, rigor was briefly mentioned in passing. Rigor was named as justification to update content-based secondary standards that "lack rigor" (Ericson et al., 2007), or in preparing students for "introductory programming courses. . . that requires rigorous logical and algorithmic skills" (Agbo et al., 2019). Other times it was an undefined requirement, like in a model proposing rigorous and coherent high school CS curriculum in Israeli high schools (Hazzan et al., 2008). In a collection of case studies in a special issue of *ACM Transactions of Computing Education (TOCE)*, Hubwieser et al. emphasized the move from current practices towards rigorous CS concepts and "rigorous academic computing" (Hubwieser et al., 2014), implying that there are rigorous and non-rigorous ways to teach CS in K-12 settings. The associated North American case study pointed out that Advanced Placement (AP) courses were considered rigorous and that students with higher scores on the AP exam were ready for college-level work (Guzdial et al., 2014). In a following special issue of ACM TOCE, Hubwieser et al. recommend factors to "implement rigorous computer science education" without defining rigor (Hubwieser et al., 2015).

In summary, from the CS field, additional conceptions of rigor are: mathematical correctness,

strictness, deep understanding of proper fundamentals, adherence to methods, formal structure of the discipline, justification for course or standard improvements, and requirements.

Prior work on rigor, in and out of computing education, demonstrates that the concept of *rigor* is highly contested, discipline-specific, and lacks consensus. Because there are so many definitions of what rigor is across educational fields, so many vague references to rigor, and because rigor is placed so centrally in many of the field's stated goals for learning and education, it is important to understand what it means in CS education, to understand if there is agreement about its meaning.

3.3 Methods for Investigating Rigor in CS education

Because we wanted to understand the nuances of the conceptions of rigor that instructors and students held, we elected to use phenomenological research methods, which would allow us to build a deep understanding of participant conceptions, descriptions, and lived experiences around a particular topic, in this case, rigor (Patton, 1999). The focus of phenomenological research is to capture personal descriptions in order to understand the experience as it is lived. We conducted semi-structured interviews to explore conceptions of rigor with instructors and students of CS across North America. We chose interviews over closed-ended surveys or statistical approaches because it allows the researchers to pursue ideas and stories that arose in conversation, and ask additional questions to better understand and to confirm what participants were saying. It also allows researchers to support triangulation of insights and support participant autonomy over how their data was represented, we also employed member checking techniques (Birt et al., 2016; Stahl & King, 2020), reaching out to each participant to confirm that the quotes we chose to represent their perspectives aligned with their understandings of rigor.

3.3.1 Recruiting

We limited participants to students, secondary teachers, and post-secondary instructors learning and teaching in the United States and Canada because this study was language-specific. Rigor as a phenomenon does not exactly translate into other languages, and since the specific word is

used so often in CS in the United States, we narrowed our scope to English instruction in North America.

For post-secondary instructors and secondary teachers, we developed an inclusion criteria that sought anyone currently teaching CS courses at the secondary or post-secondary level in North America. We excluded anyone not currently teaching CS at the secondary or post-secondary level in North America in English. To recruit instructors and teachers we used convenience sampling through the CSTA mailing list and the SIGCSE-members mailing list as well as from personal networks. Through the mailing lists, we directed potential participants to a Google form to collect contact information, experience teaching CS, and courses they were currently teaching. We selected a stratified sample of participants to ensure that the population represented a variety of secondary and post-secondary education professionals. For secondary education, we selected participants from private, public, and charter schools in both urban and rural areas. For post-secondary education, we selected participants from large public universities, liberal arts colleges, and community colleges. Additionally we used snowball sampling when participants organically recommended people in interviews.

For students, the inclusion criteria were current students who were enrolled in a CS major or a computing adjacent major, and we excluded anyone who did not take their coursework in English. To recruit students, we used personal networks to reach out through undergrad CS mailing lists at a large public university, a small liberal arts college, and a private university in an effort to find a stratified sample. We offered compensation of \$25 gift cards to each participant.

Participants

All participants self-reported gender, race/ethnicity, and any other identities they wanted to share in an optional open-field survey. Participant responses are reported in Table 3.1. If sections are blank it is because participants did not respond to that question.

Pseudonym	Gender	Race/Ethnicity	Other Identities	Major/Level	Courses
Student 1	Female	Eastern European		computer science and math	intro to object oriented programming, data structures, programming nature, scientific computing, algorithms, intro to AI, computational geometry
Student 2	Straight Man	Black and Japanese		Computer Science	Intro Series, Foundations of Computing, Intellectual Property for Engineers, Society and Technology, Systems Engineering, Compiler Construction, Hardware Software interface, Data Structures
Student 3	Male	Asian		Computer Science	Data Structure, Algorithms, Designs of Programming Languages, Discrete Math, Game Systems
Student 4				Computer Science and/or Mathematics	Object Oriented Programming, Programming Nature, Data Structures, Proofs and Fundamentals (instead of Discrete Math)
Student 5	Male	Asian	International Student	Computer Science	Data Structure, Discrete Math, and other lower-level courses...
Student 6	Female	Turkish		Biology and Computer Science	Object Oriented Programming
Student 7	Female	Asian American		COMP SCI	Intro Series, Foundations of Computing, Data Structures, Hardware Software Interface
Student 8	Male	Asian		Computer Science	Intro Series, Web Programming, Foundations of Computing, Software Design, Hardware Software Interface, Linux Tools, Databases, Data Structures
Student 9	Female	Caucasian	Disabled	CS	Intro Series (at Community College), Software Design, Data Structures, Hardware Software Interface, Interaction Programming, Linux Tools, Introduction to AI
Student 10	transgender nonbinary	Chinese American	Neurodiverse (ADHD)	Computer Science and Music Theory	Programming 1, Programming 2, Intermediate Data Programming, Foundations of Computing, Software Design, Data Structures, Systems Engineering, Hardware/Software Interface, Linux Tools
Instructor 1	Male	White		Post-Secondary	
Instructor 2	Male	White (not hispanic)		Post-Secondary	
Instructor 3	Female	caucasian		Post-Secondary	APCSA, APCSP, computer programming
Instructor 4	woman	Asian Indian/other		Post-Secondary	Java, C++, Computers & Society, Intro to Programming (Python), Programming 1, Programming 2
Instructor 5	Female	White		Post-Secondary	
Instructor 6	Male	Half Japanese, half white		Post-Secondary	Programming Languages, Algorithms (and many others less frequently)
Instructor 7	male/man	white		Post-Secondary	Many over the years, but focused on CS1 the last 7.
Instructor 8	Male	White Latino American	I am jewish, an educator ¹	Post-Secondary	CS1, Discrete math, Databases for Analytics
Instructor 9	Man	South Asian/Indian		Post-Secondary	Web development (front and back-end), mobile application developments, object-oriented programming, projects & outreach, security & privacy
Teacher 1	Female	Black		High School	Adv IT Hon and Coding Fundamentals
Teacher 2	Female	Asian	Able-bodied	High School	AP Computer Science Principles/intro to CS, AP Computer Science A, post-AP seminar
Teacher 3				All	Everything taught at Community College, run a free afterschool program for girls throughout Montana
Teacher 4	I want to identify as non-binary ²	White	No	High School	Elementary specials for grades 3-5, 3 versions of year-long intro CS, AP Computer Science Principles
Teacher 5	male	Filipino/German	Engineer/Educator	High School	AP CS-A, AP CSP, ECS
Teacher 6	Mostly Male Identified	White / Asian	Queer, Slightly Neuro-Diverse	High School	Ap Computer Science Principal, Ap Computer Science A, Game Design, Game Programming, Cyber Security, Cisco Networking, Web Design, Special Projects, Data Science
Teacher 7				High School	AP Comp Sci, Comp Sci Principles, Intro to CS, Web Design and so on
Teacher 8	male	white		High School	AP CS A, CS1 Python, CS2 Java, CS3 Java
Teacher 9	Female		Married to a man	High School	Intro, Video Game, AP CS A, AP CSP
Teacher 10	Female	White		High School	Since 1984 all grades since 2016 ECS, CSP, AP CSP, AP CS A, Advanced Topics in CS

¹Participant also included "education researcher, a computer scientist, and a software engineer"

²Participant also included "since I believe that gender is a social construct, although it feels disingenuous to identify as anything other than a man, as that is how I've lived and been treated the majority of my life"

Table 3.1: Participants self-reported gender, race/ethnicity, and other identities with the authors, and shared the major and CS courses they had taken as students or level and courses they were teaching as instructors.

3.3.2 Interviews

I conducted all the interviews over synchronous video chat (Zoom). The interviews lasted between 30 minutes to an hour and followed a semi-structured format drawing from the following questions.

- What are some things that might be important in (your) students' learning?
- When you think about CS teaching (taking CS classes), how do you think about rigor?
- How do you think your colleagues (instructors) would define rigor?
- Do you think it's possible to teach rigor?
- How do you observe rigor in students? How do you assess if students' learning is rigorous?
- Can you give me an example of rigorous work?
- Can you give me an example of work that does not demonstrate rigor?
- Could you give me an example of rigor in another domain? Would you define rigor differently in another domain?
- What would your definition of rigor be?

The questions were developed through a series of iterative pilot interviews and the first author worked with the research team to refine the questions until it was clear the participants in the pilot interviews were articulating their conceptions of rigor. Transcripts were generated by Zoom and then verified, cleaned, and de-identified by the first, second, and third authors.

3.3.3 Analysis

We adhere to Hammer and Berland's perspective on qualitative analysis (Hammer & Berland, 2014), which positions qualitative coding as generating interpretative claims about data for later testing, rather than structured data for quantification. Therefore, rather than reporting inter-rater reliability analyses and quantities, we share here our analysis process and the interpretative disagreements that emerged in building a shared interpretation.

The first, second, and third authors identified significant statements by identifying sentences, thoughts or quotes that provided insight into participants' experiences of rigor following phenomenological research principles (Patton, 1999). Since phenomenological research seeks to

study participants' lived experience of a phenomena, and find commonalities across those experiences, in this work it is important to acknowledge the embedded nature of the researchers (Emiliussen et al., 2021). The authors strove to acknowledge and "bracket" our preconceptions of the results, and our lived experiences that informed our understanding of the data as we analyzed it (Wigginton & Lafrance, 2019). After we identified the significant statements, we printed them and all authors came together to physically affinity diagram the significant statements. To do this, we grouped the statements and quotes on the walls in a large room. Once we grouped the statements, we selected a title for each grouping and inclusion as well as exclusion criteria for that group to ensure we had team consensus on what each grouping entailed. This allowed for deep discussion on interpretive disagreements.

The analysis team then reorganized and re-coded significant statements that had no group. Finally, to ensure clear groupings, we removed the title from each group of quotes. We then re-read each group of quotes to ensure that they belonged together and re-titled that grouping. In the next round of analysis, we verified the code for each significant statement. We then identified additional codes for the statements. At every step of the way, we collapsed codes until it was clear that if we collapsed these groupings further, we would lose the nuance of the conversations we had with participants.

There was significant agreement with the authors' understanding of the data. Some disagreement did arise during the initial affinity diagramming as authors debated whether participants' thoughts about effort and time should be independent categories and ultimately, collapsed them into a single effort and time grouping. Authors also debated the difference between post-secondary instructor expectations and those of state or school standards, ultimately deciding that those two entities were separate, and they emerged as independent concepts.

The final step of our analysis process was to member-check results with participants. We emailed each participant with any quotes we were attributing to them and the context in which we were sharing their quote to ensure that our analysis represented their intent. At the time of submission, the majority of participants responded (21/29), sharing that the quotes accurately represented their sentiments. Of those participants who responded, a few (4/29) requested slight

Principles of Rigor: abstract conceptions or properties of rigor	Precision	Algorithmic proof precision, clean code, mathematical exactness
	Thorough and Systemic Process	Systemic structural approach to problem solving that covered all possibilities
	Depth of Understanding	Depth of knowledge to be able to apply knowledge to novel circumstances or real life
	Challenge	Each student challenged at an appropriate level
Observations of Rigor: empirical or concrete observations	Time and Effort	Investing a lot of time or effort on a thing
	Intrinsic Drive	Self-discipline, behaviors of an individual student
	Productive Failure	Willing to fail productively and learn from it, productive struggle
	Struggle	Difficulty of material, often referred to as grind, stress, pain, or misery
	Outcomes	Grades, getting a job, knowing content
	Gatekeeping	People preventing people from learning or continuing in the discipline
Conditions for Rigor: elements necessary for rigorous learning	Expectations	Instructors expectations of student work
	Meeting Standards	State standards, curriculum standards
	Community Support	Scaffolding, online community, and friends you study with
	Resources	Money, resources, or machines

Table 3.2: Conceptions of rigor participants described, organized by broader groupings, then conceptions and a description of each conception.

grammar or punctuation clarifications.

3.4 *Conceptions of Rigor in CS*

We observed fourteen distinct conceptions of rigor, which were organized into three larger categories as seen in Table 3.2. The first category was *Principles of Rigor*, the abstract properties of rigor that participants shared. These conceptions and ideas were fairly abstract and hard for participants to concretely articulate. The second category was *Observations of Rigor*, which are

empirical observations that instructors, teachers, and students used to describe rigor, including how it manifested concretely in work, students, and other areas. The third category was *Conditions for Rigor*, which were learning conditions or environments that were necessary for rigor to exist. When we compared these conceptions across participant groups, we noticed larger trends as seen in Figure 3.1.

	Principles										Observations										Conditions													
	Student 1	Student 2	Student 3	Student 4	Student 5	Student 6	Student 7	Student 8	Student 9	Student 10	Total Student	Instructor 1	2	3	4	5	6	7	8	9	Total Instructor	Teacher 1	2	3	4	5	6	7	8	9	10	Total Teacher	Overall Total	
Principles	Precision	✓	✓	□	✓	✓	□	✓	□	✓	6	□	□	□	□	□	□	□	□	□	□	3	□	□	□	□	□	□	□	□	□	□	3	12
	Thorough & Systematic Thought	□	□	□	✓	□	□	□	□	□	2	✓	□	□	□	□	□	□	□	□	□	6	□	□	□	□	□	□	□	□	□	□	7	15
	Depth of Understanding	□	✓	□	✓	□	□	✓	□	□	5	□	✓	□	□	□	□	□	□	□	□	6	□	□	□	□	□	□	□	□	□	□	9	20
	Challenge	✓	□	□	□	□	□	□	□	✓	3	□	□	□	□	□	□	□	□	□	□	5	□	□	□	□	□	□	□	□	□	□	8	16
Observations	Time & Effort	✓	✓	✓	✓	✓	✓	✓	□	✓	9	□	□	□	□	□	□	□	□	□	□	3	□	□	□	□	□	□	□	□	□	□	5	17
	Intrinsic Drive	✓	✓	✓	✓	□	✓	✓	□	□	6	✓	□	□	□	□	□	□	□	□	4	□	□	□	□	□	□	□	□	□	□	□	7	17
	Productive Failure	□	□	□	□	□	□	□	□	□	2	□	□	□	□	□	□	□	□	□	1	□	□	□	□	□	□	□	□	□	□	□	5	8
	Struggle	✓	✓	✓	✓	✓	✓	✓	✓	✓	10	✓	□	□	□	□	□	□	□	□	4	□	□	□	□	□	□	□	□	□	□	□	8	22
	Outcomes	✓	□	✓	✓	□	□	□	✓	✓	7	✓	□	□	□	□	□	□	□	□	7	✓	□	□	□	□	□	□	□	□	□	□	7	21
	Gate Keeping	✓	✓	□	□	✓	□	□	□	✓	6	✓	□	□	□	□	□	□	□	□	4	□	□	□	□	□	□	□	□	□	□	□	2	12
Conditions	Expectations	□	✓	✓	□	✓	□	✓	□	✓	6	✓	□	□	□	□	□	□	□	□	7	□	□	□	□	□	□	□	□	□	□	□	2	15
	Standards	□	□	□	□	□	□	□	□	□	0	□	□	□	□	□	□	□	□	□	0	□	□	□	□	□	□	□	□	□	□	□	7	7
	Community Support	✓	✓	□	□	✓	✓	□	□	□	5	□	□	□	□	□	□	□	□	□	2	□	□	□	□	□	□	□	□	□	□	□	5	12
	Resources	□	✓	□	✓	□	□	□	□	□	2	✓	□	□	□	□	□	□	□	□	2	□	□	□	□	□	□	□	□	□	□	□	5	9

Figure 3.1: Each participant's conceptions of rigor sorted by type of participant. There is a heat map of popularity of conceptions at the end of each participant group and overall. Note that no two participants definitions match.

3.4.1 Principles of Rigor

When examining participant responses we observed four abstract *Principles of Rigor*.

Precision

Students referred to proofs of algorithms, mathematical CS concepts, and mathematical CS courses when they spoke about precision. In attempting to describe the abstract precision required for rigor, students mentioned examples of high-stakes algorithms and engineering processes such as bridges and planes because if the work was not rigorous, the consequences would be catastrophic. Student 2 who shared:

"Rigor is obviously something that is present in math, and is also present in programming, but it's also present in other engineering disciplines, like mechanical engineering, or even especially things like fluid dynamics. These are super complicated things that you really don't want to get wrong. You want to be able to measure even the error on these things. You know they're very rigorous. They're super thorough. You don't want a single part that is ambiguous."

Mathematical precision and exactness came up often. Both students and professors pointed to mathematical proofs and proofs of a program as a means to explain the abstract concept of rigor.

When instructors and teachers shared their perceptions of precision and thoroughness, they gave examples of code that passed tests and considered all exceptions, algorithms that were mathematically provable, and tracing program execution; where results of the precision gave exactly what was required. Instructor 6 called it the *"opposite of sloppiness,"* and Instructor 8 tied mathematical precision to the systematic thinking required to write code in CS and justified the use of autograders as a way to check for precision:

"You absolutely can teach students to do rigorous work and to value rigorous work... your code should be provably correct. You should be able to show me how it's provably correct. You should write a proof of your code. There are so many contexts where we do want that to happen... When I think about a student behaving rigorously, or writing rigorous code, I think about an engineering process that is less likely to contain mistakes about code that has been checked and rechecked, that has been tested or proven."

Completeness and thoroughness were part of *Precision* for Instructor 8. Rigor meant students were to know how to write the right code, prove it, and plan for every possible exception.

Thorough and Systematic Process

Most teachers and instructors expressed the importance of critical and systematic thought required to approach, break down, and solve problems. When instructors talked about rigor, they shared

how important it was to use systematic processes to approach a problem. Some instructors used systematic repeatedly. Instructor 9 shared:

“Rigor: meaning can they learn to systematically think through and and solve problems, so can they systematically learn to read and write code and systematically solve a problem.”

Instructor 1 expounded on that need for thoroughness and thoughtfulness when approaching problems, specifically proofs, and that one approached a proof with systematic thought, that considered all possible scenarios and did not skip possibilities or steps:

“My definition of rigor is ‘thorough’ and ‘thoughtful’ and such like that. And rigorous coursework is coursework that hits all the things it claims to hit or assesses all the things that claims to assess, or exercises in the students all the skills, that it claims to exercise.”

If students did not learn the systematic thinking to thoroughly approach a problem, they had not learned.

Teachers’ conceptions tended to focus on the steps of teaching students to think critically and independently. When teachers talked about this conception of rigor, they shared how they taught problem-solving strategies. They did this to help students learn to work independently and solve problems on their own. Teacher 6 shared the importance of teaching students to recognize the smaller components in order to solve bigger problems:

“Here’s a problem in front of me. How do I break it down into bite-sized chunks and then try to use the computer to help solve that problem?”

Only one student mentioned the process, even then as an analogy. Student 3 compared the need to shift the way they thought about approaching problems to therapy, sharing:

“I think it makes you think of therapy. It’s like reshaping the way that you think about a situation or a problem, so that you can handle it better or handle it in a different

way than you might initially have, that would have been less successful. So in that sense, I think you can also be taught to deal with rigor."

When participants described *thorough and systematic processes*, they described processes that lead to thoroughness that produced precise results; this conception seemed closely related to the previous conception of *Precision*.

Depth of Understanding

Many participants shared that rigor meant a depth of understanding in order to use that knowledge in new circumstances, or to apply it in novel circumstances. Students needed to have deep enough understanding of principles so that they could solve a problem they had never seen before. This seemed to participants to be a property inherent in some students as opposed to processes or behavior.

When students shared about rigor in CS, they did not describe the act of programming as the end objective, rather they viewed programming as a method to learn. Student 4 put this sentiment succinctly when they said, *"You learn to program, and then you program to learn."* Student 4 shared this while describing their favorite CS class. They did not learn any new programming concepts, but they were using the concepts and knowledge they had in novel ways to explore simulations of physics and in machine learning. Students often shared examples of using CS concepts to make things that they cared about *because* they had learned rigorously. They were expressing ownership of the things they were learning, rather than learning concepts because they were required to. Student 10 contrasted their opinion of learning with what they perceived professors thought learning was when they shared:

"Now that you've taken this exam you got like a 100 or whatever: that's [considered by professors to be] progress. I think it's more the purpose of gaining knowledge is so that you can use it, not to just gain the knowledge with it... If you feel that you're enabled to create something or build something, or use the knowledge that you've learned in some way."

Students described rigorous programming as a means to an end; using those skills to do something that mattered, not just for the sake of writing excellent code.

However, when instructors shared about this conception of rigor, it differed from these student conceptions. To instructors, rigor meant applying knowledge to questions that students had not yet been asked, often in the context of assessment. They shared that they wanted students to be able to apply these skills to novel questions beyond the scope or context in which they were taught. Instructor 2 shared:

"Rigor in that context would be, it's really going to be something new . . . [the solution]'s gonna have a twist [on the material] that you have maybe not seen before."

Other instructors shared that they carried a responsibility to prepare students for the next stages of life, whether that was as a software engineer, a researcher, or some entirely different pursuit, and mentioned tests or exams with novel questions as a way to ensure that students had that *Depth of Understanding*. This *Depth of Understanding* would be required for transfer, or moving knowledge to future contexts.

Teachers' conceptions seemed to fall in between instructor and student conceptions. When talking about rigor, teachers mentioned making sure students were not just following step-by-step instructions or memorizing content, but independently applying skills in new situations. Teachers shared that they found that projects were ways students could demonstrate rigor. Teacher 9 shared: *"That would be my favorite kind of rigor— it's where you're really putting it all together for yourself in a project."*

Even though the method for applying knowledge varied, students, instructors, and teachers all shared that it was important to have a *Depth of Understanding* in order to be able to use these CS skills they learned in novel settings: for instructors it was in assessments, for students it was through something they cared about, for teachers projects seemed to be the best option. *Depth of Understanding* also meant students were empowered to use these skills and take it one step further and create their own things that they cared about. Participants described a level of autonomy that allowed students to create new things. When students shared about this

conception, it was deeply tied to self-directed work and joy.

Challenge

When participants discussed *Challenge*, they focused on challenge for an individual student, sharing that what would be rigorous for one individual, might not be rigorous for another individual, but that *Challenge* was required for rigor and that level of challenge varied between students.

Students shared individual perception of levels of rigor in classes and what might be rigorous for one student might not be rigorous for another student, and that one class might be rigorous for one student and not for another student. Student 10 explained:

"I feel like rigor does vary, but I think it's very personal in how it varies. There's a lot of different factors, but I think generally it depends on the person."

Instructors shared that they needed to provide *Challenge* for their students, but that they needed to provide different levels of challenge for different students, and that rigor was not a fixed point, but was rather something that was different for each student. Reflection, and students' self-awareness were often part of the discussion of *Challenge*. Instructor 4 saw rigor as *"trying, trying trying, and pushing your limits"* and that some students would turn in completed assignments early, while other students who might have less experience, would be working just as rigorously, and would need more time and support in office hours to grasp a new concept.

Instructor 5, who had both secondary and post-secondary experience teaching CS, shared that they worked to make sure each student was able to reflect and select the level of challenge they needed:

"If I'm doing an activity in class... I am giving somewhere between two and four options of varying difficulty, so that students can practice at a level that feels comfortable, but challenging to them ... I just don't think that giving every student the same assignment could ever be rigorous for everybody."

They went on to discuss what challenging students might look like in a K-12 AP CS class, where they had witnessed teachers and instructors who would over emphasize the need for everyone to get the highest score on the end of year national AP exam. Instructor 5 then pointed out that all students would benefit from taking CS whether or not they even took the final exam.

When talking about *Challenge* and rigor, teachers focused on centering student needs in this perceptions of rigor. Teacher 6 stated:

*"I also think of rigor almost also on an individual level. Where are you at as a learner?
And what is rigorous for You may not be exactly rigorous for somebody else."*

Additionally, teachers mentioned the importance of providing multiple ways for students to engage with and represent their learning.

3.4.2 Observations of Rigor

While the conceptions of rigor in the last section appealed to abstract principles, some of the conceptions of rigor participants observed were more empirical in nature. Participants shared six such distinct conceptions. These observations came from both teachers and instructors of their students and from students of their peers. Some of these insights overlapped, but had they been collapsed, they would have lost important nuance and salience.

Time and Effort.

Often participants observed rigor as the amount of *Time and Effort* an individual would invest into learning a subject or a concept. Students', instructors', and teachers' observations of this conception varied.

Students would often discuss devotion to CS and the willingness to make sacrifices of time in order to succeed. Student 10 shared the experience of deciding when to get their work done on a weekend, and having to choose between work and family obligations when they said:

"How long do you spend thinking about the homework before you can actually do the homework? Because there's a lot of like pre-planning. I gauge rigor where it's

like like, Can you take a break from what you're doing? Do you need to be thinking about your class every single day?"

Students observed a discrepancy between the *Time and Effort* that professors anticipated would be necessary to complete assignments and their own experiences regarding the actual time and effort required. Student 9 shared some of those expectations mismatches when describing how to learn a new concept, and the professors expectations that they would invest the time to find external information to complete class work:

"I think there is a disconnect between what a professor attributes as appropriate time to spend on an assignment versus what students are actually spending on that same assignment. I've personally seen it come as a surprise when they find out that a question they thought would take students a half hour is taking students 5 hours. So they probably are thinking. 'Oh, yes, the rigor of this assignment. Yes, this is rigorous, 30 minutes', and the student is looking at this going, 'This is hour 4. I am like hitting a wall. I don't know what to do.'"

When instructors shared these observations, they focused more on the effort required to learn a new thing. Some instructors shared that they expected students to take initiative to put in effort to learning a thing. Instructor 9 shared, *"The word rigor makes me want to think about the effort that somebody would have to put in to learning or to develop something."* Other instructor shared stories of their peers who had invested into making the perfect class that would cover all the material, but that there was no way an average student would have the time to actually succeed in such a course without making significant sacrifices.

When teachers spoke about *Time and Effort*, they would refer to "hard problems," and the ways students would work with those problems, in particular the effort required to learn a new concept and engage with it. Teacher 5 compared the time and effort required to learn CS to the physical fatigue of a strenuous workout:

" That's why everybody takes computer science. They either love it because they're up for the challenge, or they hate it...You're learning like how, when you go to the

gym. You know you had a good workout... when you can't pick up the bar anymore. Picking up your gym bag to leave is like you have to drag it behind you. You know you've had a good workout. "

Students, instructors, and teachers all shared the observation that time and effort were required to learn CS, but there was a nuanced difference in their observations. Students saw the time and sacrifice required to do work well and instructors and teachers focused on the effort required to grapple with a new concept.

Intrinsic Drive.

Many participants described that they observed an intrinsic quality, habits, or mindset that students either had or did not have.

Students shared observations of their peers who were always focused and studying, spending all their time reviewing course material, or on researching additional content. Student 7 laid out two ways they observed rigorous students:

"There's two ways to get good. . . One is, you're really good at it. Like you're just naturally good. Two is, you work really hard. Kind of try to apply those to computer science or kind of in school in general. . . I really rely on the second one. I think I'm afraid I'm not that talented at programming. I've relied a lot of the kind of like just grinding."

Student 7 put themselves in that second category, but perceived that what required much work from them came naturally to a lot of their peers. Other students used language like *"strict and exact"* when observing their rigorous peers' behaviors, or *"passionate"* when observing their rigorous peers' mindsets.

Instructors' observations were split. Some instructors said rigor could not be about students or their qualities, but rather about the complexity of the course. Few instructors thought students attitudes could be rigorous.

Many teachers expressed that they observed rigor to be the drive students brought to the work they did. Several teachers used expressions like, *"extra mile"*, *"discipline"*, or *"pushing one's self"*. Many teachers perceived some students to be rigorous and some students to not be rigorous. Teacher 4 speculated that they thought rigor and intrinsic motivation were correlated, sharing:

"Perhaps I associate rigor with intrinsic motivation... I guess maybe if you're intrinsically motivated, you want to do it for the sake of itself. And so you want a challenge, and you want to learn and get better."

Some teachers perceived some students and their academic behaviors to be rigorous— while other students did not practice the same behaviors.

Productive failure

Participants shared they observed rigor when they watched how students recovered from failure in coursework and in learning.

When students talked about failure, they did not focus on the failure, instead they focused on how good it felt to get it right after the failure. Several students also surfaced the tension they experienced that they may not have the space to fail in class. Grades often played a role in the fear of failure— which made it hard for students to take some of the risks they felt were necessary for learning CS. Student 10 shared the analogy of building a sandcastle:

"I feel like not being stressed is one of the important things, because I feel like learning should be more of a sandbox environment where you can do what you want... If you're building a sand castle, you wouldn't have someone yell at you because it didn't work the first time that you tried to build the sand castle. And so it's the idea of being able to experiment when you learn and also it's not gonna be like. 'Oh, I'm gonna fail you in the class, because, you messed up your first assignment.'"

Few instructors shared that they thought failure was part of rigor. Instructor 3 shared the importance of the reflective process when failing rigorously:

"Rigor includes challenging oneself to do something difficult. And we're not always going to succeed. So rigor also includes experiencing failure, reflecting on what can be done differently, and trying again."

When teachers emphasized the importance of failure in rigorous learning, they often included the importance of community support, like working with peers or with a teacher to find mistakes and bugs, and that the school system was not always designed to support that collaborative nature of failure. Teacher 3 was emphasizing how important it was for everyone to learn, not just elite students who would end up in college or with PhDs, when they said:

"You have to learn certain things, and you don't have the ability to fail like you do when you're in school, and that's the big... readjust from those mistakes and move forward... especially in computer science, because that's what it's all about. You're failing. You're constantly failing. You constantly have bugs, and you constantly made a mistake somewhere. And you're always trying to fix these mistakes."

Teachers saw the constant iteration on buggy code as a necessary part of rigorous learning.

Struggle

When talking about this observation, students and instructors shared different understandings. Students' perceptions were often negative while instructors shared positive perceptions of the struggle.

Students often used words like *"suffering," "miserable," "grind,"* and *"unhappy"* when describing what was required to be successful in a course. Often CS students would describe long hours and in front of a program or in a library, and the sacrifice of a social life or sleep in order to be fully dedicated to finishing an assignment or being a good student. They equated the state of being stressed out with the level of rigor in a course. When asked how they knew a course

was rigorous, Student 10 replied, *Stress. Being stressed out. . . But I think stress is probably the main way that rigor is present, or that I can feel it at least.* And Student 9 pointed to the difference in perception between students and instructors by sharing an example of the contrasts in how professors perceived rigor and struggle and students did:

"I think sometimes the rigor can be discouraging or straight up, prohibitively difficult. I think that sometimes some assignments and things like that are just straight up prohibitively difficult in how they're laid out...And you'll end up seeing things like a professor announcing, 'Hey, so I saw enough of you cheated on the midterm that I'm making a public post about it,' and I think that that's an indication that the rigor may be a little bit too high. If people are feeling like they have to resort to something to be able to even pass."

They then shared that when they observed their peers cheating, it was as a last resort to not fail a class, not due to laziness. They thought that this was a sign that the course was too hard.

Many instructors had a very different perception of this concept and used words like challenging and struggle with a positive view. Although, some instructors reflected about colleagues who made things hard to make things hard. Instructor 3 who shared the tension of creating courses that were challenging because they covered the necessary content, but not difficult just to be difficult:

"I think that some would say, 'Rigor is making students do hard things.' And I don't think I would disagree with that. But I also think faculty need to deliberately avoid creating assignments that are difficult for the sake of being difficult. What is the learning objective? How can we remove any cognitive load that isn't related to solving the problem?"

Sharing a similar reflection, Instructor 8 spoke about a colleague who was dedicated to providing thorough course content, and would try to fit as much as possible into a term. As a result, students were overwhelmed.

When teachers talked about rigor as a part of course difficulty, they often started by stating what work they thought was not rigorous; e.g. multiple-choice, reading a book and bubbling in answers. And when teachers described rigorous work they used words like *"hard"* and *"struggle"* often with positive connotations. Many teachers thought it was how hard a course was, like teacher 5 who said, *"Rigor is the level of difficulty."* Then they went on to share ways they supported students to succeed in spite of the difficulty.

Outcomes

Participants mentioned outcomes as a way they were able to observe rigor in themselves (as students) and their students (as instructors). Surprisingly, participants rarely mentioned grades as a goal of outcomes, but did mention things like *"doing well"*, and *"testing"*. Students saw outcomes like gaining content knowledge, reflecting on the new things they were able to do, and passing tests. Student 1 initially referred to rigorous students as doing well in the class, and later corrected herself, *"Rigor means, practice and practice means result, and then I [thought] 'you know, rigor doesn't always mean good grades.'"* Student 4 also shared similar thoughts differentiating between rigor and succeeding in a course:

"I think the classmates that do well are more rigorous. But this implication doesn't necessarily go both ways. If you do well, you're rigorous. But if you don't do well it doesn't always mean you're not rigorous."

Many instructors shared that they more concerned with outcomes like mastery or making sure students knew the material well. Instructor 4 said, *It's our job to get [students] to demonstrate that competence and to be able to feel confident in what they're doing.* When instructors mentioned grades, it was in the context of confusion around what they communicated to students and about students. Instructor 5 pointed out how grades and identity were entangled, and mused:

"When we talk about rigor and teachers who maybe think about rigor in a more traditional sense. 'Everybody has to achieve the same level of difficulty in order to be

A level or B level... ' I wish that our education system was a little bit more flexible in terms of how we think about students, and not thinking about it as like. This is like A level, and if you're not achieving A level, then you're not an A student.'

In general, instructors shared that they wanted to make sure that they were able to help students master content so that students were prepared for careers in CS.

In addition to observations of outcomes like mastery that instructors held, teachers often tied rigorous outcomes to projects that worked, and that students could clearly explain. Teacher 8 mentioned how they knew students had rigorous outcomes and really understood when they explain why broken code did not work.

Gatekeeping.

Often students shared stories of having to fit in, or meeting perceptions of others in order to survive in CS. It seemed that in order to be rigorous, or perceived as rigorous, students shared that they felt like they needed to behave like their peers. Student 5 clearly links fitting in to a performance of rigor when they shared:

"It can prove to be really difficult to survive in this environment... If you don't fit, well, everyone is all about fitting, and everyone is all about the rigor. So everything in the program is about rigor. So it's really difficult to navigate this environment if you're not what they want or what the standard is."

Several instructors shared things they had witnessed in staff meetings, conversations they had been in, observations of peers, or personal stories of ways that students were discouraged from pursuing CS because they were not *rigorous* enough. Instructor 3 shared one such anecdote about her son:

"I have a son who is taking a computer science class at a college right now, his first one, and he says, 'Mom, the teacher makes me feel like an idiot.' And I think that happens a lot in computer science and some people call it rigor. It's not."

Instructor 1 shared that they wrestled with the word *rigor*— as they felt it had some value or proxy for professional decision making about students. But when reflecting they articulated:

“Rigor is gatekeeping. It’s the word we use when we want to pretend there’s some actual measurement behind our gut instinct for whether people are succeeding honestly or not honestly, like cheating. But, honestly, like at a level we consider acceptable. Maybe that’s it. I want to use that word but... I wanted to have some actual concrete definition. I just can’t find it. ”

3.4.3 Conditions for Rigor

Whereas the first and second sets of conceptions of rigor were about the behavior of teachers and students, a third set of conceptions concerned the properties of systems, such as assessment policies, standards, and community resources. This third set is not a set of definitions, but rather a set of necessary conditions teachers, instructors, and students saw as necessary for rigor to exist.

Expectations

This condition for rigor was about instructor expectations for their students. It primarily came from post-secondary students and instructors. Instructors worked to ensure that students met expectations, while students shared they often struggled to understand what those expectations were.

Many students mentioned the need to meet instructors’ expectations in order to succeed in post-secondary CS courses. Students expressed frustration that they did not always know what professors wanted, and that they were doing many things to try and meet those standards. In trying to describe professors’ expectations, Student 9 shared an analogy:

“They show you a picture of a chicken, and then they tell you to draw a duck. So you’re never gonna get it right. But you’re gonna get some of this stuff right, unless you’ve seen a duck. But they don’t show you a duck, so you’ll never know. They’ll

say it goes quack. You're like cool. Let me guess what that means. But there's not enough information to be able to get it right the first time... The level of rigor of a course is always going to be viewed completely differently from like a perspective of a professor/teaching team and the perspective of students taking the course. and it makes it kind of difficult to understand before going into a class, how difficult it's going to be."

Other students expressed similar frustrations and experiences of trying to figure out if the professor was withholding concept details because that was the best way to learn, or if a professor was unaware that many students needed more instruction to feel successful.

Instructors saw their own expectations as benchmarks for student success. If students were able to meet their expectations, they were learning and succeeding. Several instructors acknowledged that while clear expectations promote student success, they also feared overly high standards might make classes too difficult and hinder student achievement. Instructor 2 reflected on rigor when designing courses:

"Students will often go to the expectations we set for them... If we set really high expectations for what we expect them to do, they will a lot of times meet that rigorous demand on them... We don't want to make classes hard for the sake of them being difficult or for creating a reputation of 'This is the class where you learn rigor.' But there is value to having high expectations... I would hesitate to apply rigor to students rather than to the course."

Instructor 2 felt it was important to delineate between courses and students. Courses could be rigorous, test students over novel content, and have high expectations. However, they did not want to apply the concept of rigor to students. Expectations served as the marker by which students were assessed in undergraduate and graduate CS programs. In secondary classes, the vocabulary differed slightly: teachers often used the language of *Standards*.

Standards

Middle and high school teachers relied on standards as a way to observe and verify that their students were rigorously learning. In this sections, we refer to standards as school, district, or state standards that teachers were charged to help students meet. Only teachers mentioned this observation of rigor.

Teacher 1 shared the importance of really understanding what the standards were, and how to teach them in order to make classes rigorous. They shared that standards ensured rigor in their classroom:

"I envisioned rigor to be teaching to the standards and then using cumulative assessment rather than evaluative assessment to address a particular objective or problem within the course of study."

Teacher 6 shared how difficult it was to teach CS because students come into the class with different levels of experience, and different needs in the classroom, but that all students needed to meet the standards by the end of the year:

"Students might need different things in order to get to that standard. And the way different students show that standard might not be exactly the same."

Teachers viewed their ability to help students meet standards as a way to benchmark whether or not they were teaching rigorously.

Community Support

Many participants shared that in order to have rigorous learning, it was important to have peers, mentors, and faculty who supported them. They felt rigor did not happen as well in isolation. Students and teachers mentioned these ideas much more often than college and university instructors did.

Students shared stories of making friends with online moderators who taught them tricks about writing code, going to office hours with professors and TAs, and studying with friends. Student 1 shared the importance of community and peers they learned CS rigorously:

"There will be people grouped together writing it out on boards. We're doing homework together. Because, like I said before, rigor is hard to learn that on your own, [unless] you are guided through, maybe by someone who's better at it, or someone who's also learning."

When talking about supporting students, teachers often talked about scaffolding and meeting students at their ability level. They shared stories of individual conversations they had with students to assess those abilities. They talked about the care required to not just answer a question but how to help a student figure it out and develop a deeper understanding of CS. This work went beyond teaching just content, and meant teachers and community members were investing in students and relationships to support students. Teacher 1 shared about the support they needed to provide their students:

"Rigor has to be relevant. It has to be scaffolded for whatever learner that is present in your class. I can't be just teaching to the top ten or twenty percent, but about the students who fall in the middle, what about other students who, you know, need additional support."

Similarly, Teacher 8 shared that you could not separate rigor from meeting a student where they were to empower them, which require teachers to know each student well enough to know where each individual student was in their level of understanding as well as their emotional state. They thought that a classroom community was required to do so when they shared:

"I think sort of rigor in teaching is being systematic about what information students need, at what time, in order to feel empowered and approach new challenges creatively without becoming overwhelmed."

Resources

Resources like access, time, money, and computers were elements that were part of rigorous learning. Students, instructors, and teachers all shared examples of systemic barriers that were tied to a lack of resources that slowed them down or prevented them from learning and teaching rigorously.

When teachers spoke about *Resources* needed to support rigorous learning, They quickly pointed out lack of resources like time and money. Teacher 6 expanded on how these resources interfered with teachers' ability to teach rigorously, sharing:

"I think that they think that rigor is like making sure that students are challenged appropriately. I think they may have a harder time of essentially quantifying what that is for making sure that it's being applied to everybody. And I think part of [what makes that hard] is also just systemic barriers or systemic racism and ableism, and all that sort of stuff like we're in a system that kind of automatically. It pushes people out of rigor."

One instructor mentioned that due to limited public school resources and lack of access some students came in with no exposure to CS, pointing out that it was not quite fair to require the same level of rigor as those who had been programming for years. Reflecting on the societal inequities that slowly built up over the course of his schooling experiences, Student 2 shared that they did not have CS courses in their high school before sharing:

"There's all sorts of directions or sources of rigor that kind of create this web of... Not only is it academic, but it's also societal. There's like a huge huge web that feeds on itself on just multiple sources of little little sources and big sources of rigor that create this almost perfect storm of being able to have such a rigorous system in place."

Other students shared how they had the resources that enabled early access to CS through parents, friends, or games that allowed them to pursue CS rigorously from an early age, and

recognized that as a privilege.

3.5 Discussion

Our results show that over the course of twenty-nine interviews, 14 conceptions of rigor and no clear consensus on the definition of rigor in CS education emerged. Participants discussed many different **Principles of Rigor** which included *Precision*, *Systematic Thought Process*, *Depth of Understanding*, and *Challenge*. Additionally, participants shared **Observations of Rigor** which included *Time and Effort*, *Intrinsic Drive*, *Productive Failure*, *Struggle*, *Outcomes*, and *Gatekeeping*. Finally, participants shared **Conditions for Rigor** which included *Expectation*, *Meeting Standards*, *Community Support*, and *Resources*. No two participants' definitions of rigor matched, as seen in Figure 3.1. Each student, teacher, and instructor we spoke with had a different definition of rigor in the discipline of CS. Instructor 2 pointed out that they observed this confusion in conversations with faculty peers:

"Rigor is one of these interesting words, right, because people will mean different things when they say it. . . the word gets mixed up with with a lot of other things."

Many participants shared that they did not hold a simple definition of rigor, and on instinct, asked to look it up, so as not to give a wrong answer. But, in this study we sought to understand what conceptions of rigor participants held and had observed. Some conceptions of rigor were about teacher behavior, some about student behavior, other conceptions were about student work products, and student knowledge. While it may not have been clear what participants thought rigor was, it is clear that there was not one definition of rigor, and that an individual's role as student, teacher, or instructor impacted their conceptions of rigor.

In previous work examining rigor, we know of disciplinary specific conceptions from the liberal arts and STEM. In this work, all of the conceptions presented were rooted in CS. To our knowledge, this work is the first seeking to understand what conceptions educators and students hold of rigor with respect to CS. As Riley notes, it is important to understand what each discipline assumes rigor to be (Riley, 2017).

A careful examination of Figure 3.1 illustrates some of the differences in the understandings of rigor between participant groups. Although there are not exact boundaries, there are some observable trends. Instructors and teachers held more conceptions of *Principles* than students did. Students held strong conceptions of the *Observations of Rigor*, particularly *Time and Effort* and *Struggle*. In fact, each student shared *Struggle* as a definition of rigor. This difference in definitions between instructors and students may mean that the field may not be clearly articulating what rigor is to students.

When discussing **Principles of Rigor**, The concept of *Precision* matched with mathematical conceptions of rigor and with conceptions of correctness (Ginat, 2007a; Wohlwend & Pepler, 2015). Participants often shared high-stakes scenarios where human lives were at stake. When participants spoke about *Systematic Thought Process* and *Depth of Understanding* they mentioned thoroughness as part of those two conceptions. Thinking through things thoroughly was part of a systematic approach. The concept of *Depth of Understanding* supported work by Ramnath et al. that rigor and relevance were important in student work (Ramnath & Hoover, 2016). Participants shared joy when they talked about the ability to apply knowledge to new circumstances, or when they understood how larger parts tied together. There was a particular pride that participants evoked when they shared about using their knowledge to create new programs, games or things that they cared about.

When sharing **Observations of Rigor**, participants' concepts of *Time and Effort* aligned with theoretical frameworks that perceive rigor as hard work or time (Campbell & Dortch, 2018; Culver et al., 2021; Draeger et al., 2013, 2015). Students shared examples of their misery and grind culture when trying to put in the *Time and Effort* they felt necessary to master material. Teachers and instructors seemed less aware of the sacrifices students were required to make outside of class in order to learn rigorously. Additionally, a few teachers and instructors seemed to perceive *Intrinsic Drive* almost as a personality trait that some people were born with. But, it is important to consider that study skills are learned behaviors, and that not all students have the same levels of executive function (the ability to plan, set goals, and focus) and that sometimes policies can conflate behavior and worth (Kirdani-Ryan & Ko, 2024).

The concept of *Productive Failure* was difficult for some students because they saw the importance of failure, but felt limited by grades because failure meant lower grades. Instructors hold power over the grades, as they assign them, and did not mention anything about the conflict between *Productive Failure* and grades. We have designed education systems for success, not for failure. But, by its very nature, CS learning requires failure. For example, debugging is a process that is often explicitly taught because code does not work and fails and must be debugged. Our educational systems of deadlines and grades are not always designed to support the iterative process of such learning.

With respect to *Struggle*, there is a broader conversation about power, where instructors might be creating assignments that are "hard," where difficulty without appropriate support and acknowledgment of students' personal constraints (e.g. familial responsibility, financial needs) that affect a student's ability to invest enough *Time and Effort* to meet an instructor's goals—where *Struggle* becomes *Gatekeeping*. In classes, instructors hold a tremendous amount of power over students' experience, outcomes, and futures (Everson et al., 2022).

Computing teachers and instructors also risk entangling student identity with *Outcomes* like grades, or behaviors like *Intrinsic Drive* or *Productive Failure*. If classroom instructors make assumptions about students' ability to learn based on instructors' own perceptions of what rigor is, they risk excluding students who do not have the same lived experiences as instructors. As instructors, we know what it took for us to be successful, and we might recreate these scenarios—but really, what we're recreating is success for people who have the same life circumstances we do. This risks excluding anyone who holds different identities and or backgrounds.

When discussing **Conditions for Rigor**, participants' ideas of *Expectations* align with the theoretical framework of cognitive challenge to students (Campbell & Dortch, 2018; Culver et al., 2021). Instructors perceived their expectations as a way to push students. However, it is important to make *Expectations* explicit—often, students shared confusion and frustration about what professors wanted them to do, and they would waste time and effort in an attempt to meet unstated expectations.

Community Support and *Resources* highlight educational inequities. Students shared examples

of times where they had access to CS in elementary school— which ignited their desire to study CS. Other students shared that their high schools did not offer CS, so the first time they were introduced to computational concepts was in a freshman CS course, among peers who had been practicing these concepts their entire lives. Yet, both types of students were expected to succeed in the same *rigorous* course.

This paper found, similar to the work done by Draeger et al., that there is a mismatch between faculty perception of rigor and what faculty communicate to students (Draeger et al., 2015). In CS in particular, professors are hoping to articulate precision, and students internalize rigor as the need to grind and make sacrifices in order to prioritize CS. Our results not only show that there is no agreement in CS, but that this lack of shared agreement has impacts on how students are trying to learn. Throughout all of these conceptions, we heard participants' stories of teachers and instructors taking a particular notion of rigor legitimacy and applying it to the students they are teaching— legitimizing particular conceptions and behaviors.

3.5.1 *Our Understanding of Rigor*

Rigor is a combination of discordant conceptions educators use to try to ensure students are learning. Educators may believe that rigor arises when students learn difficult things so they use difficult coursework as an aspirational goal for their materials. When rigor is the justification for learning, it legitimizes behaviors that are traditionally dominant, conflates meritocracy and dominance, and continue to perpetuate the white masculine behaviors that many students do not perform (Riley, 2017). Slaton and Riley argue that, "fortitude and suffering are required for success"— which contributes to the continuation of stratification and violence within the field (Slaton & Riley, 2024). To survive this use of rigor, one must conform and conforming reproduces the initial conditions of CS which excluded so many (B. J. DiSalvo et al., 2011)— which may explain why the field continues to struggle with representation (Jane Margolis et al., 2017).

The use of the concept of rigor perpetuates the current white masculine hegemonic state of CS. Rigor is in part about the adherence to the discipline and precision (Riley, 2017). Adherence to the traditions of a discipline means adherence to how it is presented, how it is taught, and

how it is perceived by students. Students already come with hegemonic perceptions of tools. Introducing new contradictory experiences (e.g. sewn circuits when teaching circuitry) broadens the perspectives that students hold, and students learn the content better (Peppler & Thompson, 2024). Until instructors present new tools, ways of knowing, new forms of pedagogy, and content in CS, the field perpetuates exclusive traditions and expectations (Cheryan & Markus, 2020).

Epistemologically, if CS education uses the concept of rigor to legitimize the discipline, we are perpetuating the traditions and expectations that already overrun the field. In doing so, researchers and practitioners neglect other ways of knowing that could benefit both students in their learning and the field for growth and exploration (Turkle & Papert, 1990). Centering epistemic narrowness and a cost-effective transfer of content becomes the focus of CS education rather than knowledge and learning (Shank, 2000). When rigor is centered, CS education risks limiting not only who joins the field, but also the capabilities of the field itself to take risks, to try new things, and to construct new knowledge.

Of course many of the concepts like *Precision*, *Systematic Thought*, and *Depth of Understanding* surfaced in this study are essential to CS. We should not abandon such ideas. However, the term *rigor* is not clearly defined, and reinforces homogeneity which in turn harms the field.

3.5.2 *Moving Past Rigor*

There are many options to replace rigor, one argument might be to clearly define rigor for CS, so that each instructor and student is clear about what rigor is, and how to be a rigorous student. Alternatively, computing could entirely reject the concept of rigor in CS education. The first step would be to admit that rigor is not as objective as the field perceives it to be. In acknowledging its subjectivity and the arbitrary definitions we give to rigor, only then can CS education begin to construct an understanding of what would benefit student learning (Harding, 1992).

What replaces rigor might make learning hard to control, and it might make it harder to measure, stratify, and rank students. It would require CS education to focus on and prioritize student learning over content presentation. It would require instructors to build trust with students and believe that when students are entering the learning environment, students are there to

learn and that students want to succeed (Stommel et al., 2020). It would require systems and institutions to better support instructors and teachers— so that teachers and instructors have the support, resources, and time to build that trust and new curricula. This type of learning would be difficult to scale, and CS education would require more labor of instructors than systems currently allow. For example, instead of 200-student lectures, it might be 20-student classrooms, where relationships, humanity, and trust can flourish and where students are able to create and engage in the joy of learning hard things. We see shadows of this joy in conceptions of rigor in *Depth of Understanding* where students know material so well they own their knowledge and use it as a tool to create things they care about. To do this it might require instructors and institutions to incorporate student voice when creating assignments that matter and defining learning outcomes. It would require an examination of power in the classroom, reconsidering how grades are earned and assigned so that courses are not about competition, but about learning. We also see it in *Productive Failure* when students finally get the code to compile, or finally debug that final flaw, when they have space to make mistakes and try again. Instructors would need to consider how to allow for students to fail before they succeed, failure takes time, and often current course timelines only allow for success.

3.5.3 Limitations

This study has several limitations. This study was done in North America with English speakers, as the authors spoke English, however this study due to its methods, excluded many perceptions of rigor around the world, and limited the conceptions of rigor to ones that were part of North American academic culture.

Our results also have methodological limitations. Participants volunteered for this study. We recruited them through education specific listservs. All of the teachers and instructors interviewed were generous with their time and thoughts. They cared about their students and the courses they taught and they each shared how they were working to be excellent educators. This is a threat to this study's results as being a quality educator was part of these participants' identities. Additionally, students were all enrolled in CS or adjacent majors, meaning the recruiting methods

excluded any students who had self-selected out of CS, had failed out, or had left the CS major.

Additionally, results were also limited to the viewpoints and perspectives of these participants, if other participants had been selected, it is possible that additional or alternate conceptions of rigor would have been identified. And although researchers worked hard to bracket their assumptions, interpretations were potentially limited by these researchers' experiences and opinions.

3.5.4 *Implications*

This study extends our understanding of rigor particularly as it relates to CS. The participants considered rigor an abstract concept and articulated that it is hard to know whether or not students are rigorously learning, or applying rigor to their work. Because it is abstract, CS education relies on observations of rigor. The need to perform rigor shows up explicitly in the ideas of *Gatekeeping* where students spoke about the need to fit in or behave like other students behaved which reinforces hegemonic norms as Riley argues (Riley, 2017). CS educators share opinions of performing rigor so that their knowledge, identity, and occupation are legitimized. Perhaps, like Nelson mentions, these legitimized opinions prevent excellent CS teaching (C. E. Nelson, 2010).

This study finds that, particularly in CS, no one person can know all of CS, and yet often designers of courses and assignments expect that of students. Educators should be clear about what it means to thoroughly know a topic in CS and work to clearly articulate expectations of students, and work even harder to help students meet those clearly defined expectations.

High school teachers are putting forward tremendous effort to teach CS rigorously. But it is clear the field has vast and varied views of what rigor is. This work calls into question learning standards, and what notions of rigor are implicit in our assessments, and what those assessments means. Additionally, this points to a major hole in CS teacher education at all levels— as teachers can not articulate what excellent CS instruction looks like. Finally, this work raises significant concerns about how students come to see rigor, through lenses of suffering and sacrifice.

3.6 Conclusion

This work finds that no two people have the same definitions of what rigor is. Often, rigor is used to decide if students are *good enough* to be computer scientists. Instead, we should discard the concept of rigor as it applies to students, and focus on whether the instruction we are providing is *good enough* to support ALL of our students. We need to explicitly communicate our expectations, standards, and what it takes to succeed to students, and we need to work to make sure that each of our students has the resources to meet those standards. If we create a new definition of rigor, it runs a very high risk of becoming a new rigid structure that we use to decide whether students are *good enough* rather than if we are providing quality education. The goal of computing education should be to help students learn. Student success should be at the center. Students are capable of doing hard things, and when they do those hard things we ask of them, we should celebrate that success.

Chapter 4

STUDENT DREAMS OF JOYFUL LEARNING: DREAMING OF DIFFERENCE



This work was written with Rotem Landesman as co-first author, and with the analysis support of F. Megumi Kivuva, University of Washington's Youth Advisory Board and Amy J. Ko. Work is always done in community, as such in this chapter uses the term "we" to describe this research team. A version of this work was accepted and presented at ACM's 2025 International Computing Education Research Conference (Everson et al., 2025).

4.1 Why do we need to ask students what they want?

Since rigor is often used as justification to make pedagogical decisions designed to gatekeep students, I wanted to know what pedagogy high school students wanted out of CS classrooms. Students are so rarely asked what they need. Rather, they are often told what is good for them. However, in my experience teaching, I've found asking students what they need, means that I can better support their learning, and that as a team we can together begin to shift the agency

of learning from the teacher to the student. I wanted to know what students found helpful in their learning and what they dreamed of if they were put in charge and could have any learning experience in the world.

As CS is being recognized as an important component of secondary education around the world, the number of CS classrooms is increasing. With that rise in CS classrooms comes an urgency in training teachers to fill those roles, efforts to create infrastructure, the creation and amendment of learning standards, and finding of sources of funding for these efforts (Code.org, 2022; DeLyser et al., 2019). Due to this rapid expansion, researchers are exploring what CS education should look like in secondary classrooms.

Prior work suggests that demystifying CS while emphasizing the social and societal impacts it has on students can support engagement in CS classrooms (J. J. Ryoo, 2019b). Others note the importance of adapting CS learning materials to particular students in order to build confidence and strong relationships (Hwang et al., 2023), prioritizing student autonomy and collaboration to build strong identities for students in CS (S. L. Johnson et al., 2025), and using multiple conceptions of CS topics to help students build stronger understandings (Margulieux et al., 2021). Researchers have also looked at learning barriers in CS classrooms, including those to effective learning (Dong et al., 2021; Lewis, 2023; G. L. Nelson et al., 2017; Yadav & Berges, 2019), enrollment (Bean et al., 2025; Childs et al., 2024; Ladner & Stefik, 2017; Rankin et al., 2021a, 2021b), and factors that contribute to students' lack of interest in CS content (Childs et al., 2024; de Wit et al., 2023; Downes & Looker, 2011; Happe et al., 2021).

All of these prior explorations of what CS education should look like are critical as we continue our efforts to expand and diversify who can and wants to learn CS. However, most of this prior work reflects what researchers, educators, and other adults think CS education should be, rather than what the students themselves think. Research that has been done with young people often sought their opinions regarding the content of CS education, rather than the pedagogies educators should use in CS classrooms. Rolden et al., for example, worked with teens to co-design informal learning spaces in libraries, and found ways in which students could build sustaining relationships, disrupt power, and challenge computing norms (Roldan et al., 2022). Similarly, Ryoo and Blunt sought

to examine Latine and Black youth's perceptions of CS to see how well those perceptions aligned with culturally responsive computing pedagogy. They found that students believed CS is political and held power, schools should empower students to be responsible creators of technology, and that addressing the social implications of technology would attract diverse learners (J. J. Ryoo & Blunt, 2024). Both of these studies surfaced students' views on the broader power structures present in classrooms and the content that should be included and highlighted in CS learning environments.

While prior work has built a strong case for using particular theoretical framings of CS pedagogy, which this work defines as “classroom teaching practices” (J. J. Ryoo, 2019b), we do not yet know what students desire CS pedagogy to be. Soliciting youth voices regarding what pedagogies we use in CS education is key to learners enrolling and thriving in CS. While surfacing those perspectives on pedagogy, it is equally important to solicit students' current perceptions about CS in order to situate our understanding of their desired pedagogy. Doing so may give educators and researchers insight about how they make decisions about enrolling in computing classes (J. J. Ryoo & Blunt, 2024; Wang et al., 2016) and how to build CS pedagogies, content, and environments which are inviting to secondary students.

We ran 5 co-design sessions with secondary students to solicit their current perceptions of CS instruction, and collaboratively dream of the future of what CS education should be, seeking to answer the questions:

- **RQ1** What perceptions do secondary students have of CS pedagogy?
- **RQ2** What pedagogical desires do secondary students have for CS education?

4.2 Prior Work

4.2.1 Student perceptions of computing

Prior work has explored students' varying perceptions about who does computing and what computing is. Through a large survey scale phone survey, Wang et al. found that students perceived that smart white men were the people doing computing, and that computing included

things like making presentations and creating documents. Students in this survey also thought that boys would be far more likely to be both interested in and successful in learning CS, and that they needed to be both smart and good at math to succeed in CS (Wang et al., 2016). Other work shows that most students did not see themselves as likely to have a job needing CS (Wang et al., 2017). Building on Wang et al.'s work, Childs (Childs et al., 2024) found that students often equate CS to only coding, while an interview study with teens based in the United States (US) found that even though teens were actively engaged online through a variety of platforms and tools, they had little interest in CS as a subject or career and perceived it to be difficult, boring, and asocial (Yardi & Bruckman, 2007).

Students' negative attitudes towards computing mirror their perceptions of careers which involve computing. A study in Ireland with teens found that they generally perceived a career in computing as boring and hard, where they would have to sit with computers all day. These students voiced these views although they had prior exposure to computing, which the majority reported as a good experience (Nolan et al., 2023). Similarly, work done by Rouvrais and Kanellos in an introductory computing course in France found that students left secondary education with negative views of computing that did not change even though the course's goal was to help students build a more objective view of CS (Rouvrais & Kanellos, 2011).

Scholarship exploring students' goals when enrolling in CS programs has mostly examined the perspectives of students who are taking or have taken secondary computing courses. In their work, Mark et al. surveyed New York City students about their reasons for enrolling in Advanced Placement Computer Science Principles, a course designed to broaden participation in computing in the US. They found that young women, Black students, and Latine students were placed in the course, whereas young men often elected to take the course on their own (Mark et al., 2021). In other work, Ryoo et al. interviewed students to understand perceptions that students held about CS broadly, and found that students saw computing as political and powerful and therefore task teachers with preparing students to be socially responsible creators (J. J. Ryoo & Blunt, 2024). This prior work provide us with a greater understanding of what students perceive as the overarching goals of computing education. We still do not know, however, how secondary

students believe CS should be taught. Their views will provide researchers and educators with the knowledge to structure the pedagogy of computing courses in a manner that recognizes the perceptions of secondary students who have yet to take these classes, which may boost enrollment.

4.2.2 Designing CS learning with secondary students

Though learning has been shown to be more effective when it is personally meaningful (Priniski et al., 2018), it can be difficult to identify what is interesting or meaningful to young people (B. DiSalvo & DiSalvo, 2014). To uncover what is personally meaningful, researchers have turned to designing pedagogies and learning environments with teens, since designing this way “offers opportunities to uncover self-motivations, identities, and interests and to construct meaningful engagements by working together with students to co-creating learning environments that meet the needs of the whole community that is engaged in the learning practice” (B. DiSalvo & DiSalvo, 2014). Though there are many approaches to design with teens, among them co-design (Druin, 2002), Participatory Design (PD) (B. DiSalvo & DiSalvo, 2014), and Youth Participatory Action Research (YPAR) (Ozer et al., 2020), all of them equally value placing teens voices at the center of the conversation, and allow teens to express their need and wants through the design process. Designing with teens allows students to leverage their funds of identity, knowledge from a lived experience in the world, to build a new group identity where they are able to value their peers and their peers’ ideas (Iriarte et al., 2023). In addition, designing in collaboration with youth also allows them to reflect on their identities, think about their future selves, and apply those identities to their designs (Coenraad et al., 2019).

Researchers have explored many topics by designing with youth. For example, Cicuto et al. co-designed with attention deficit/hyperactivity disorder children using physical computing to provide new means of communication, expression, and agency (Cicuto et al., 2024). Co-design has also been used in a variety of instructional endeavors, teaching computational thinking (Sunday et al., 2024, 2025) and learning Scratch (Morales-Navarro et al., 2022).

A few prior works have explored youth co-design of CS learning spaces, revealing some youth preferences. In work done by Roldan et al. (Roldan et al., 2022) teens led co-design workshop in

libraries to re-imagine informal computing education experiences that would serve their communities. This work pointed to relationships as underpinning the work teens did as they disrupted traditional notions of power and acted as experts in these design sessions (Roldan et al., 2022). Alternatively, Chang et al. (Chang et al., 2024) used Youth Participatory Action Design with historically marginalized youth to re-imagine and dream about AI technology. Their work found that comfort and vulnerability were essential in these interactions, and that youth wanted AI to support effective learning and safety (Chang et al., 2024). These prior works, however, have yet to design with teens in an effort to uncover the pedagogical characteristics they see as important in secondary CS classrooms. Doing so will allow researchers and educators to elevate students' voices and lived experiences, and integrate them into CS pedagogies in future and current classrooms.

4.2.3 *Speculative Design*

Some prior works have used speculative design methods to imagine possible futures related to computing. Speculative design is a methodology that seeks to imagine what is preferable in all of the probable, plausible, or possible futures. It is a way to imagine alternative realities and alternative futures, helping us to understand and design for futures without the cultural norms or harms that currently exist. It can be done through storytelling, brainstorming, design, and art (Dunne, 2013). Researchers have used speculative design to re-imagine the cultural contexts of technology (Nias & Holmes, 2023), to re-story dominant narratives in computing with Black girls (Shaw, 2023; Shaw et al., 2023), or design social media for and with Black women (Klassen & Fiesler, 2023). Others used it as a method to imagine a decolonized future for CS education with indigenous populations (Kareta et al., 2023), or to support ethical speculations as a skill to imagine all the possible harms that might come from current technologies (Botes & Rossi, 2024; Fiesler, 2021; Klassen & Fiesler, 2022). Additionally, when used with children, speculative design enabled adults to support critical reflection about AI (Malinverni et al., 2025). Speculative design provides a way for students to dream of alternatives to traditional CS classrooms.

4.3 *Methods for Dreaming of CS Classrooms*

Building on prior work, we chose to co-design with teens because we contend that they are the experts of their experiences, and that their voices, which are informed by their experiences and identities, are essential to listen to and to value when building new educational spaces. We also chose to use speculative design with secondary students to imagine CS education opportunities and possibilities that were not constrained to current conceptions of school and CS classes. With this in mind, our work contributes the voices and dreams of what CS pedagogy can and should be in secondary education informed by youth.

4.3.1 *Students and Contexts*

We conducted this study over the course of four virtual workshops and one in person workshop with teens (ages 14-17), exploring how they perceive and envision the future of CS education. Each workshop was 2 hours long, and included adult researchers (the two first authors) as organizers and students (N=18) described in Table 4.1. The group of teens who participated were members in the authors' university-affiliated youth advisory board (YAB). YAB is a group of students who meet twice a week in-person over the summer, and once every two weeks remotely during the school year. Once a semester throughout the school year, the group meets in person to participate in activities, catch up, and discuss relevant topics. Members of the YAB work with researchers, graduate students, and undergraduates on research projects and local advocacy initiatives. Participants in the the YAB have relationships with each other given their sustained and long-term work together, and with one of the first authors who leads the group throughout the year. This allows for honest and open discussions on the topics the YAB explores in the course of the year. To mitigate organizers' influence over students' responses, many of the activities described below begin with students reflecting in small groups in breakout rooms independent of organizers. Students were recruited from local high schools by snowball sampling and word of mouth, and met throughout January and February of 2025 to conduct the workshops described below. Though the study was conducted through the University of Washington, a university lo-

cated in a region known for its technological advancement, the teens recruited were not sampled for any specific prior knowledge of CS, their experience with CS related curricula, or their identities, but instead chosen because of their eagerness to learn and to share their lived experiences. Our participants attended seven high schools across the region. The schools represented here have free and reduced lunch program participation rate between 12-27%, which is lower than the national average of 50% and the state average of about 40%. In the United States, free and reduced lunch participation rate is used to understand school funding and socio-economic status (Domina et al., 2018). As we did not select these participants for their racial or gender identities and did not use these identity as a lens to analyze this data, we chose not to include race and gender demographics as recommended by Oleson and Xie et al. (Oleson et al., 2022).

Participant Pseudonym	Age	Workshops Attended
YAB_A	17	W1, W2, W3
YAB_B	14	W1, W2, W3, W4
YAB_C	16	W2, W3, W4
YAB_D	17	W1
YAB_E	17	W1, W2, W3, W4, W5
mikrokosmos	16	W1, W2, W3, W4
RoboGeek24	15	W1, W2, W4
YAB_I	17	W2, W4
YAB_J	15	W1, W4
lloveChEeSe	14	W1, W2, W3, W4
Coolcat13	15	W3, W4, W5
YAB_M	17	W1, W2, W4
YAB_N	15	W1, W2, W3, W4
Cat_lover-824	15	W1, W2, W3
YAB_P	14	W1, W2
white butterfly	15	W1, W2, W3, W4, W5
YAB_T	14	W1, W2, W4, W5
Simi	16	W1, W3, W4

Table 4.1: Participants' self-disclosed demographic information

This work intentionally aimed to work with students who had a variety of previous CS experi-

ences, ranging from no formal experience to enrolled in a CS course. This is in contrast to prior work by Ryoo and Blunt (J. J. Ryoo & Blunt, 2024) who specifically interviewed students after they completed CS courses, meaning those students had concrete experience that informed their perceptions of CS. The state in which the study was conducted does not require CS classes to graduate with a secondary degree, and each of the schools the participants attend offered elective CS classes. Out of the 18 participants in our workshops, 3 reported to have taken CS elective classes in their secondary coursework. The voices represented in our work are also important, and provide valuable insight into the vision prospective CS students have about CS education. Attendance of students varied throughout the workshops, and is indicated in Table 4.1. We refer to all students' by pseudonyms, some of which they chose themselves. If students did not choose a pseudonym, we assigned them a random letter (e.g. YAB_Z).

4.3.2 Position Statement

As researchers, we recognize the need to address our own agendas and views in research, when working with youth and sharing their perspectives. The authors are researchers and educators with over 30 collective years of formal and informal teaching experiences at the secondary and post-secondary levels, with varying levels of expertise in computing disciplines. With an underlying commitment to equity in education, we approach this work with critical perspectives and are deeply committed to making sure youth voices are heard, especially in the context of computing and education. This played out in sessions by researchers consciously refraining from sharing their own opinions. Additionally, we did not discuss the concept of pedagogy or pedagogical theory with students, or frame the questions we asked students as inquiring about pedagogy. Rather, we asked questions about what students wanted their CS educational environments to be.

4.3.3 Design Sessions

The first four workshops were loosely scaffolded to engage students' curiosities about research and and introduce them to CS education research. The final design session focused on member-

checking (Birt et al., 2016) the major themes the researchers surfaced with the teens and integrating their feedback into this work's final coding scheme. The sessions were centered around exploratory activities and open-ended discussions, and aimed to support and surface teens' experiences and perspectives rather than fulfill the goals of the researchers. In addition, since the workshops were offered as part of the YAB group, each session introduced and let teens practice one or more research methodologies, such as thematic qualitative analysis or design based research ("Design-Based Research", 2003). Each of the five design sessions began with introductions and an icebreaker (approximately 5 minutes) to build rapport and ensure continuity from one session to the next. Each session then offered one or two main activities, with group-wide discussions following each activity (approximately 45 minutes each). At the end of each session, teens completed individual research journals (approximately 15 minutes) to reflect on the session's activities and discussions, which were used for data analysis.

Workshop 1 (W1): Evaluating Curricula and Platforms.

The first workshop focused on exploring existing platforms and programs for teaching CS. The session began with a small group activity about who should assess a CS learning platform when it enters an educational environment, their roles in the school ecosystem, and the metrics they might use to evaluate said platform. Then, the students were asked to share their reflections with the larger group, and collectively think of assessment criteria for a CS learning platform. In the second half of the session, students were asked to choose one or more common existing CS learning platforms (such as Roblox ¹, or Scratch ²) and assess that platform with their criteria, while thinking about further modifications and clarifications they would like to make to the assessment. At the end of the session, students were asked to fill in their virtual research journals, reflecting on the activities and discussions in the workshop.

¹<https://www.roblox.com/>

²<https://scratch.mit.edu/>

Workshop 2 (W2): Design a CS Learning Environment.

The second workshop used speculative design to ask teens to imagine an ideal CS learning environment, as well as an ideal CS learning curriculum. The first activity asked students to split up into small groups and create a visual representation of their team's ideal learning environment, considering the important aspects of the space in which their learning would take place. By using digital images, colors, and words, the groups each made a 'vision board' of an ideal CS learning environment (Figure 4.1), and then shared their vision with the larger group, promoting discussions about the different elements students chose to include in their designs. Then, students were asked to go back into their small working groups and imagine characteristics of ideal CS learning by reflecting on guiding questions such as: What does the learning look like? What do the projects or assignments or homework look like? What kinds of support do you have? What roles do adults play? What roles do students have? After brainstorming in their small groups, students were invited to share their insights with the larger group, where we discussed the criteria of learning, drawing on the teens' lived experiences. Finally, students were asked to reflect on the discussions and activities in their virtual research journals.

Workshop 3 (W3): Qualitative Data Analysis.

The third workshop was designed to engage students in qualitative thematic analysis (Braun & Clarke, 2006), as both a learning opportunity about the methodology and an exercise in identifying the themes they themselves surfaced during the first two sessions. This workshop began by collaboratively reviewing one of the author's prior scholarly works that used thematic analysis as its methodology, to demonstrate the manner in which thematic analysis showed up in research. Then, students received a spreadsheet with all of the research journal entries they've submitted over the previous two sessions, with all responses anonymized so that they could not recognize the author of each entry. In small groups, teens used sticky notes on a Miro board (affinity diagram (Okoko et al., 2023) the anonymized journal entries into themes using sticky notes, and then shared and finalized the themes surfaced in the larger group format. After discussing the

themes and settling on their titles and descriptions, students were asked to fill out their research journals and reflect on the workshop's activities.

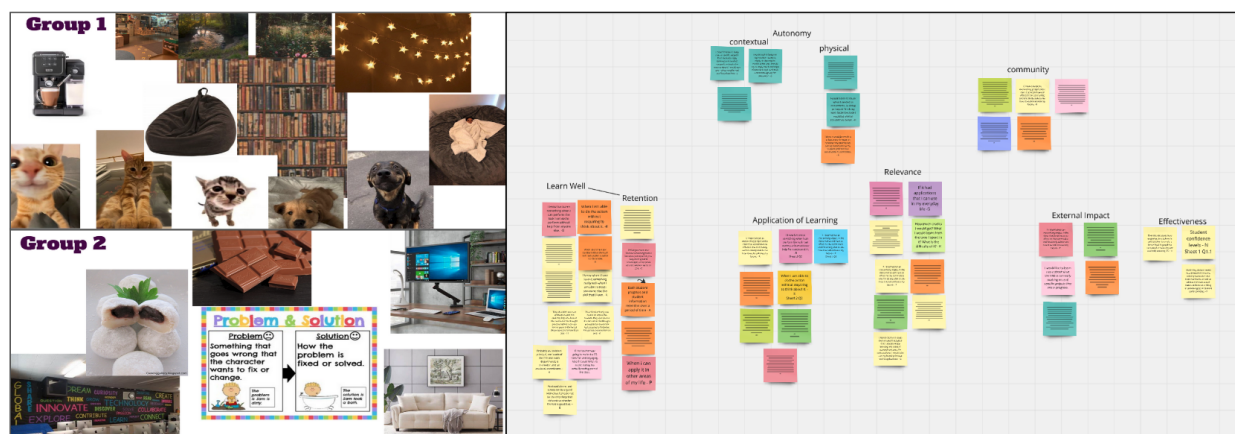


Figure 4.1: **Vision boards and student analysis.** On the left are some of the vision boards students created in workshop 2. On the right is the affinity diagramming board that students and an organizer constructed during workshop 3.

Workshop 4 (W4): Dissemination of Research.

The fourth workshop engaged students in discussions and activities about dissemination of academic research. We discussed two dissemination techniques selected by the organizers, after students asked how research was shared: writing letters to policy makers who can use findings from research studies to change public policy, and creating a research poster to present a study's findings to a larger audience. In the first part of the workshop, students worked in small groups to outline a letter to their state senator, organizing what they would like their CS education to look like. For the most part, students looked at their previous analysis and research journals in order to complete the outline, and then shared their initial thoughts with the larger group. For the second half of the workshop, students drafted a similarly positioned research poster about their insights into the potential of CS education, and then shared their design posters with the larger group. Finally, students were asked to reflect in their research journals about the day's activities.

Workshop 5 (W5): Member Checking.

The goal of the last and fifth workshop was to triangulate our analysis and findings with the students themselves through a process of synthesized member checking. Synthesized member checking (Birt et al., 2016) is a process where synthesized, analyzed data is returned to students to provide them with an opportunity to comment, amend, and add insights to the findings. The member checking process involved several key steps: After the authors finished analyzing the data and agreed on a coding scheme (detailed in Section 4.3.4), we presented the code book, including code definitions, sub-codes, and anonymized example quotes to the students. Then, we asked students to see if they could recognize their own experiences within the synthesized themes, and solicited their thoughts, reactions, disagreements, and contentions with our findings. Students agreed with and supported most of the themes the research team presented, making a few suggestions to further nuance the findings for this study (detailed below in 4.3.4).

4.3.4 Data Analysis

Our data sources were primarily workshop recordings and student artifacts. We recorded and transcribed all workshop discussions, collecting a total of 13 hours of audio. Additionally, we included teens' written research journal responses as well as artifacts from the group activities across all sessions in our data corpus.

We used thematic analysis to qualitatively analyze the data (Braun & Clarke, 2006, 2021) in this study. The first phase was done with students during W3, so as to center their voices and understand their intents. The second phase started after the workshops had ended and all data was collected. The first three authors started with a line by line open-coding of the data to surface any significant statements (Braun & Clarke, 2021), identifying codes such as "good learning = teaching others" and "CS class is worth time = relevance". Following this initial coding phase, the authors collaboratively discussed their observations and grouped the codes by students' perceptions of CS, and students' claims about what CS education should be. Following Hammer and Berland's recommendations to view qualitative thematic analysis as interpretative

claims about data (Hammer & Berland, 2014), we resolved disagreements by building consensus through discussion. For example, one point of disagreement among the authors which required clarification by discussion was teasing out the specific ways in which students advocated for their peers in their visions of an ideal learning environment, and to what extent that advocacy was clear through an 'as-is' reading of the data, as we sought to minimize inference in our analysis.

The authors then collaboratively created an initial codebook with codes such as "Community/Collaboration" and "very mathy-y", which they then applied to the significant statements they originally pulled out of the data to check for compatibility. Throughout this process, the team iteratively refined the codes, discussed any disagreements (as described above), and adjusted the codebook as necessary to produce the final codebook with 12 final codes. Six codes answer RQ1 and reflect *students' perceptions of CS education*. The remaining 6 codes answer RQ2 and address *what CS education should be*. The second set of codes had additional sub codes and can be found in Table 5.2. At least two authors coded every piece of data and documented their disagreements.

After the authors agreed on a coding scheme, the third author went back and coded the entirety of the data. Only then did the team examine their thematic analysis against that of the students (which they completed in W3), and noted the differences and similarities between the two, amending codes like "autonomy/contextual" from the students' analysis into "autonomy/-choice" in our final code book. The authors then brought the final code book to the students and engaged in member checking in W5 (see 4.3.3). We then verified that students felt comfortable with the codebook and the representation of their ideas, as well as collectively reflected on any points of tensions in the codebook as we presented it. For example, students reiterated the difference between the necessity of CS education to be engaging, but not necessarily fun because that felt frivolous, and did not capture their value for learning. This prompted the research team to amend that code and refine its definition. Following that session, the research team met to finalize their codebook and discuss the final results presented below.

Codes	Description	Sub-Codes
Responsibility and Accountability	Participants envisioned CS classrooms as having a distributed model of responsibility and accountability among the students themselves, the teachers, and classmates. Holding each unit accountable for fulfilling its responsibilities would encourage effective learning	Teacher responsibility & accountability
		Student responsibility & accountability
		Class responsibility & accountability
Community Oriented	Participants highlighted the importance of community in CS education, both in the sense of making collaboration an integral part of the learning process and engaging with communities outside the classroom to ground their CS education	Collaboration inside the classroom
		Collaboration outside the classroom by means of work making an impact and teaching others CS content
Value Learners as Complex Individuals	CS education should value learners as holistic and complex individuals who come to CS learning with different backgrounds, lived experiences, and different needs. With that understanding, CS education should strive to provide each learner what they need to grow best	
Autonomy	Participants expressed a need for autonomy in their CS education	Participants expressed a need for autonomy in their CS education
		Bodily autonomy
Engaging and Relevant	CS content should be relevant to learners' lives and engaging in its teaching practices	Engaging
		Relevance on either a personal level (relating to a future career or hobby) or on a communal level (creating a lasting impact with the work)
Support Learning	CS education should strive to support meaningful and long lasting learning	Confidence
		Able to teach it
		Able to apply it

Table 4.2: Final codebook for *What CS education should be* from thematic analysis

4.4 *Student's Dreams of CS*

We structure our findings by first describing what students perceived CS education to be (RQ1), and only then describing what these same students think CS education should be (RQ2). By doing this, we aim to showcase the contrast we saw between students' current perceptions about CS education and their visions of its future, which we expand upon in the discussion section.

4.4.1 *Perceptions of CS Education*

CS is Math-y. Students consistently brought up notions that CS, and any of its related content, was very heavily informed by mathematical knowledge and theory. While this is not an untrue assumption of CS in the way it is often taught, we saw students wanting to avoid CS because being a math-y person was not an identity they held. This sentiment and identity conflict was reflected by several students. For example, when YAB_C was asked if they had tried coding they said:

"Yeah, I'm in [a course], right now, which is computer science principles. I feel like it's a little bit too math heavy for me. Personally, I'm not very good at math, and it's very hard."

CS is Hard. Students shared that they thought CS is a hard and a difficult subject to study, reserved only for highly talented students. YAB_M shared:

"I think that like CS classes in general just kind of have, like a bad reputation, maybe for being boring and not really engaging, or just really like content heavy and hard."

Students generally described CS classes with words like "dread" and "difficult." They also expressed that they worried about not being able to keep up with the work.

CS is Boring. Students perceived CS and its concepts are boring and irrelevant to the work they wanted to do. YAB_T quoted something they heard a classmate say, *"ugh I have to take [a CS class] next period it's SO BORING!"* As students' perceptions of CS are formed by their

peers' experiences of CS courses, sometimes that meant they were not hearing anything about these classes, when they did, it was exclusively negative.

CS is Individual. Students thought that learning CS and the profession of CS were individual endeavor, in which students would spend the majority of their time working independently in competition with each other rather than collaboratively. When asked what collaboration would look like in a CS class, student YAB_T shared how it could be different:

"Although learning CS seems to have a reputation of being taught in schools as students working on a monitor separately, there are many ways to innovate and make it so it's more fun and effective, especially in schools."

CS is a Black Box. Students shared that CS classes were an unknown and therefore taking a CS class is a risk. Coolcat13 was recently registering for courses and said, *"I didn't even know that there like was computer science options available until I started signing up [for next year's courses]."* They also shared stories of other classes which they were excited to take because they saw the projects and the impacts those projects had on their school community, unlike CS classes which had no visible projects or impacts.

CS is a "Golden Ticket." Students saw learning CS as a means to a lucrative career path that they did not need if they were not considering it as a career. YAB_J shared about peers enrolled in CS who were taking it because, *"it's just something they think they might have to do in the future."* Additionally, they shared that they perceived taking CS courses was a way to distinguish themselves when applying for college. YAB_E reflected about friends who were enrolling in CS, because *"they want to take those classes that are like known for being really hard, so it'll look good on college transcripts or college applications,"* stating that CS seemed like a way for students to stand out to colleges and universities at application time.

4.4.2 What CS Education Should Be

Throughout the workshop activities, students surfaced six elements of an ideal vision of CS education and how it should be taught, which we describe and elaborate on in the sections below.

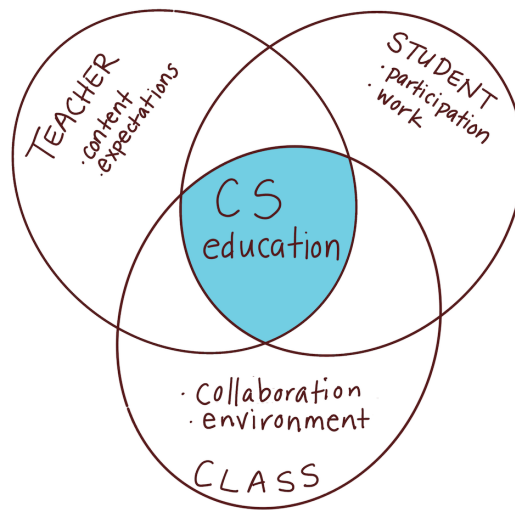


Figure 4.2: **Responsibility** the interdependent model of responsibility and accountability and that students described.

CS Education should have responsibility and accountability.

Students envisioned CS classrooms as having a distributed model of responsibilities and accountabilities among three entities: the individual students themselves, teachers, and the class as a collective. They saw each entity as having its own set of responsibilities which they are held accountable to fulfill, and depend on the other two to uphold in order to foster an environment of learning and collaboration as pictured in Figure 4.2.

Teacher responsibility and accountability. Students named teachers as responsible for making the content in the CS classroom "engaging, but still informative and educational", as well as "making sure people with different learning styles are able to like, keep up" (YAB_T). Students noted that teachers are responsible for "[having the] relevant experience in both CS and teaching", further adding that "they should also be willing to understand and take feedback from students" (YAB_B). The responsibility of teachers to set clear standards for success in the classroom, as well as hold students accountable for meeting those standards was a especially salient among students, like YAB_A who explained that:

"I mean ideally, [classroom standards] could be built by both the teacher and student, but most of it comes from the teacher. If the teacher is, I don't want to say too soft, but I have a teacher like that, who [sees] the students are unmotivated, [so] he sort of becomes that way as well, and he doesn't really do anything to discipline them. And so the class just becomes unenjoyable."

Student responsibility and accountability. Students reflected that they themselves, as active students in a CS classroom, would have a set of responsibilities they'd be accountable for. For example, students would be responsible for *"evaluating the clarity and usability of the [CS learning] program"* (YAB_A), and *"provide more specific case-study type information that reflects how a diverse group of students benefits or falls behind [when engaging in a classroom's] learning format."* (YAB_E). Youth viewed students as also responsible for fostering a collaborative classroom environment by *"taking a minute to like, connect with your peers as well as like work on work on whatever you need to do for the class."* (YAB_C), as well as upholding standards set by the teacher for classroom behavior, deadlines, and responsible use of digital tools.

Collective class responsibility and accountability. Finally, students referred to the collaborative classroom environment itself as accountable for providing them the resources and support they needed to learn CS. While engaging in speculative design activities in W2 especially, students dreamt of a CS classroom that allowed them and their classmates to use multiple methods of learning by integrating *"elevated monitors"* which provide *"lots of space to write, which makes it easier to take notes"* (YAB_P), but not foregoing an environment which centers collaboration, as YAB_A explained:

"[in regards to] the tiered [seating], that's what happens in my in my school's CS lab, and I think that works pretty well because when the teacher's presenting, and the monitors get pretty bulky, but you still want to see everyone and interact with people. I think it's helpful when it's in rows. And so that even the people at the top can see and interact with everything that's going down below. it's kind of like amphitheater style."

Many of the more cosmetic environment elements students suggested, like comfortable chairs (because *"just coding nonstop for 90 minutes and staring at a screen on one of those plastic or metal chairs is super uncomfortable, and if you take CS for several years your your back is gonna start to deteriorate."* (YAB_A)) or having available snacks in the classroom (*"because snacks can help you focus, especially if you're hungry like, while you're studying."* (YAB_P)) embodied the commitments they believed the environment should make to all its inhabitants to foster learning — to be comfortable, inviting, and considerate of the long term effects on each of its students. While these elements can be an after thought in the grand scheme of CS education, students expressed just how much they wanted to see their classroom environment reflect that they cared about each other while also reflecting their values and learning goals. YAB_A, for example, explained that though she thought the learning environment should be comfortable, it should not be *"too relaxed, [as there's] a balance between that and also educational."* Students recognized the responsibility of a classroom lays, then, in advocating for each other, and making sure everyone's needs were met.

CS Education should be community oriented.

Another facet of CS education students surfaced is the importance of orienting around community, both in the sense of making collaboration an integral part of the learning process and engaging with communities outside the classroom to ground their CS education.

Collaboration inside the classroom. Throughout the workshops, students consistently brought up collaboration with their peers, whether that be through *"making it a more collaborative environment on projects or like on homework. . . . Just taking a minute to connect with your peers as well as like work on [class]work"* (YAB_C), as *"collaboration is important. Just like how like, we're collaborating here just like kind of building off each other because it creates new ideas and just makes for like a better like classroom environment."* (white butterfly). Students offered practical ways of increasing collaboration in the classroom, by having *"students face eachother"* (YAB_E) or *"if we're doing a project making it a group project instead of an individual one, or like finding like random fun ways to like connect with your peers instead of just, you know. . .*

working on your coding the entire time" (YAB_C).

Collaboration outside the classroom: teaching others and impact. Students also surfaced the importance of collaboration outside of the classroom. One way of giving back to the communities they are a part of and encouraging collaboration, for example, may be by teaching younger students part of what they learn in a CS class. While sharing out her group's design of a CS classroom to the entire group, YAB_A explained that:

"[Our group would] incorporate as much collaboration and project-making [as we can], whether that be through teaching younger students, or collaborating with local companies (this is something that (anonymized) high school did over the pandemic. . . . I think it's not only beneficial to both students and and whoever they're helping, but it also allows to see the end result of all the learning that they're doing and the potential for how their knowledge can be used in the real world. Which is super motivating, I think."

By teaching others what they've learned and making sure their work can impact their communities, the students highlighted the importance of collaboration that surpasses the walls of the CS class.

CS Education should value learners as complex individuals.

Furthermore, students noted that CS education should value learners as holistic and complex individuals who come to the learning space with different backgrounds, lived experiences, and different needs. With that understanding, students reflected, CS education should strive to provide each learner what they need to learn and grow best. Foregrounding students' voices, opinions, and goals proved to be incredibly important to students who reflected that:

I think, just emphasizing the [importance of] student voice, and even the families like their goals. And I think, like, when you design a [CS] assessment, thinking about what students goals are, what they want to get out of that course, and then like

checking in like throughout the course, to see if they still feel like that goal is realistic and something they can like feasibly achieve is important. (Simi)

Subtle ways to make students feel capable in their learning can go a long way in showing that educators value the student not just in their role as learners, but as capable and constantly changing human beings who need to be reassured, inspired, and confident when they approach new topics. White butterfly explained that *"in school, there's a lot of times where, like you get stuck, and it's really helpful if there's like a poster on the wall, or something that tells you how like other people might have solved it. . . that'll help you figure out how to solve your own problem."*, underscoring the importance of acknowledging students' desires to feel confident and independent in their learning. This notion of recognizing students as nuanced and complex individuals extended to recognizing their unique ways of learning. As YAB_T explained, they *"sometimes get the urge to learn and experience as much as I can. If I could structure how I learn, I would work in pomodoro breaks. I would also focus mostly on learning during the early morning (I think that I work best then), and following productivity curves while maintaining discipline. I would learn many subjects - not just what schools commonly offer."* If CS education can embrace the unique complexity of each of its secondary students, these students believed, we would be able to achieve deeper and more meaningful CS learning experiences in the long run.

CS Education should have autonomy.

Students expressed a need for autonomy in their CS education, mentioning both autonomy in choosing the topics they learn and bodily autonomy (the ability to move their body as they needed) in the CS learning environments.

Choice of Topics. Students wanted to be able to choose how and what they were learning. YAB_E shared the they wanted to learn, *" topics that would be more relevant to you and to your life... and in actually being able to learn the subjects for a long time,"* and that the ability to dig in and really understand something was an important part of their learning. mikrokosmos shared that if they were in charge of their learning they *"would choose to deep dive in specific subjects*

that I actually enjoy learning and conduct research on topics I'm curious about." In addition they felt it important to be able to choose to *"use my learning for real world applications."* These students were highlighting the importance of choosing work that they were curious about, and that mattered to them. When students talked about autonomy, there was often an overlap with their desire for learning to be engaging and relevant (see section 4.4.2).

The desire to choose work that was interesting was often talked about in contrast to current classes. YAB_C reflected that when they wrote, *"A lot of my classes right now are useless and I end up just doing [other] work during them"*. Students also found it important to acknowledge that they really did want to understand the essentials when they were learning things they cared about. YAB_N shared: *"I would make sure I have the fundamentals down in the topics that are overall important and then I would choose new things I'm interested to learn about after that."* This highlights an interesting tension where students are not experiencing the fundamentals they are learning as relevant to what they want to do.

Bodily Autonomy. When students were designing the physical spaces they wanted to learn in, they mentioned comfort, calm, and collaboration as important functions of those locations. For example, since school is not an optional activity, and the chairs students sit in at school are uncomfortable, students were seeking alternative setups, where they could choose how to move and what was best for their bodies while learning. YAB_P shared with their discussion group,

"[we] wanted like a comfy area. So that's why we put a couch. Because, sometimes it's really hard to sit at a desk for like 8 hours a day, so like working on a couch can be really comfortable."

In addition to comfortable seating, students highlighted how stressful school was, and the importance of finding ways to contrast that stress, like mikrokosmos's group who worked to curate an environment that had:

"a lot of cats and dogs to like make the environment more relaxing and like, get rid of stress... and also we put nature there because we like wanted to go outside and take breaks often."

Students recognized the importance of a calm and relaxed environment and its impact on their learning. With the caveat, as discussed in section 4.4.2 that this calm and relaxed environment was also one where teachers held students accountable for their work and their actions.

CS education should be engaging and relevant.

Students routinely expressed that CS education should be relevant to learners' lives and engaging in its teaching practices.

Engaging. When students talked about engaging courses they often mentioned that they wanted to learn CS as it related to other subjects they were already learning or interested in, in ways they could immediately use it. YAB_B shared that they would be eager to take CS if they were able to spend time, *"learning skills that are relevant to other fields I'd be interested in."* They reflected that they were not interested in CS as a career path, but they wanted to know how they could use it in other subjects that they were considering. And another student shared that they wanted CS class to be, *"fun and engaging"*, and expressed a desire to enjoy their learning.

Doing relevant work that matters. It was important to students that CS education should be relevant to what they are were interested in and their lives. We heard students articulate this in two distinct ways: first as it applied to their personal goals and hobbies, and second as it applied to their greater community, to make an impact.

Matters personally: relevant to career goals or hobbies. Throughout the discussions and reflections, students expressed that they wanted to study CS if it would be relevant to their personal lives. YAB_E shared,

"[I would take CS] if it had relevance to my life or interests. I do want to take a CS class to have a better understanding of how computers and websites work, as well as just understanding code and modifications to websites or games. A lot of the time, I think connecting subjects to others is a very good way to both get motivated to learn things and fully understand things."

This reflection points to an their underlying belief that CS courses as they are are not helping

students achieve the goal of understanding how CS is entangled in their lives and studies.

Other students shared that they were not planning on pursuing CS as a career, so they were not interested in taking CS at all because they were not convinced it would be useful. For example, YAB_I wrote that they would take CS if they could have *"a guaranteed stat showing that it's better to take CS in today's world, [that] would probably convince me to take it."*

Matters to the community: makes an impact. Students repeatedly emphasized how they wanted to do work that mattered to the communities and the worlds they live in. They want to do work that matters and they want to do work that is worthwhile. White butterfly wanted to take a CS class that, *"would have an effect on the community,"* and that they wanted classes where they were able to, *"learn something big that like makes a difference, or just like making a difference in general... just like some way to make a difference like in the community, or the classroom, or something."* Student perceived much of their learning as divorced from 'real work' that mattered, and expressed a desire for classes where they could have an impact with their time, effort, and energy.

CS education should support learning.

Students saw worthwhile learning in CS as learning that inspires confidence, allows learners to teach content to others, and inspires learners to apply their newly acquired knowledge to other areas.

Confidence. Students were very thoughtful about how they knew they had learned something. Often they reflected that confidence was one way they knew. YAB_T explained the nuance between getting something right, and understanding it when they said, *"sometimes, I'll get the answer right without actually understanding anything. So I also think, like taking students like confidence levels like on a scale throughout the lessons, it's like important."*

Teach others. When students felt confident with their knowledge of a topic or a subject, they also felt they were able to teach it. YAB_M shared:

"I find it really helpful like, after I've learned it... to teach it to a friend or someone,

just because I feel like it gives you a lot of confidence that you really know it enough to be able to like share it with someone."

The ability to teach it to someone else was also a way that students knew they had mastery over a topic, like YAB_T who shared *"being able to explain to others such as friends, family, and especially younger children/people who have no knowledge in a field is a huge indicator to me that I really know a subject."*

Apply it. The third way that students were able to recognize their learning was when they were able to apply the content they were learning. White butterfly shared how when they could apply it in other areas of their life, it positively affected those other areas of their life as well, reflecting, *"I think also like the application, just being able to apply it to so many things which then, I think, inspires other people. And it kind of just creates kind of like a ripple effect, but also to just better understand."* YAB_J pointed out the independence that goes hand in hand with knowing and applying what you have learned when they said, *"just like feeling good about what you learned, and knowing that you can like, apply it to other things without maybe the help of your teacher, or like other people."*

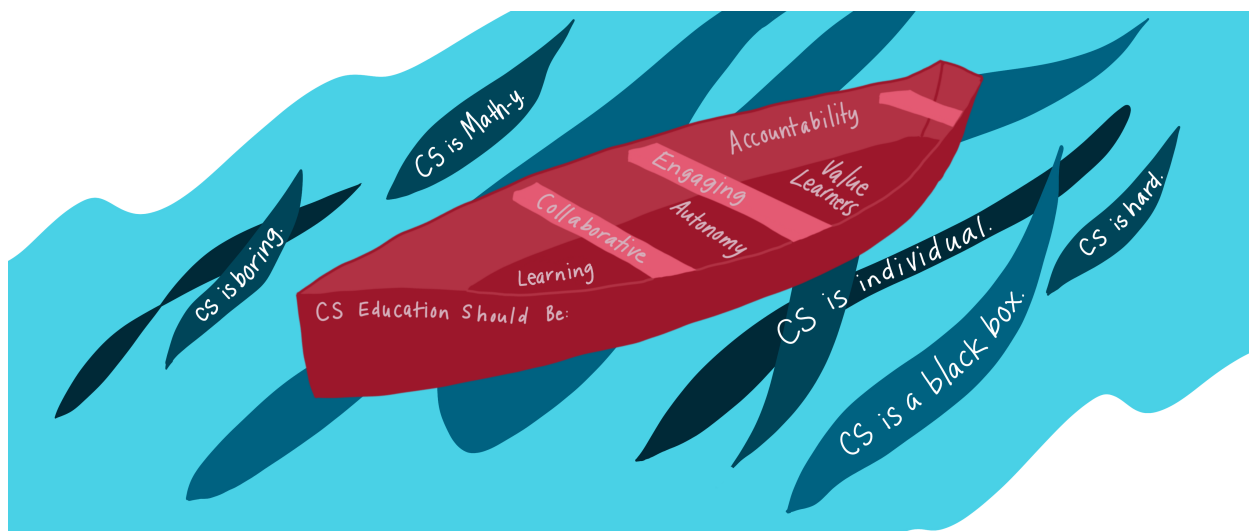


Figure 4.3: A boat filled with the students' desired attributes of a CS classroom against the current of a river full of students' CS perception.

4.5 Discussion

This study explored what pedagogy secondary students desire from their CS educations, as well as what they perceive current CS education to be. Through a series of co-design sessions, students articulated their perspectives on the importance of foregrounding distributed responsibilities and accountability, ensuring CS education to be community oriented, recognizing the complexity and nuances in themselves and their peers, valuing autonomy as well as making sure classes are engaging, relevant, and promote learning. These visions of the CS education and pedagogy students want to experience and participate in are contrasted with their current perceptions of CS education, which they viewed as boring, math centered, difficult to learn, a black box with bad optics, and a sort of "golden ticket" to future careers and opportunities. By surfacing these perceptions and wishes, this work paves the way for educators and researchers alike to reflect on our practices in the CS education space, and collaboratively re-imagine with youth what future endeavors may look like.

We found that these students' perceptions of computing aligned with the previous studies by Wang and Yardi, that CS is math-y, hard, and boring (Wang et al., 2016; Yardi & Bruckman, 2007). They also perceived it as an individual exercise where they would code with a computer all day which aligns with Nolan's work (Nolan et al., 2023). Students thought that CS was mainly for people who wanted to major in it as a career choice, as found in previous literature (Wang et al., 2017). In this study, we found an additional contribution that students perceived CS classes as 'black boxes' because they didn't know what happened in them, and signing up for a CS class would be a risk, as they might end up in a boring unenjoyable class. These perceptions students hold are in stark contrast to the types of CS classes that they want to take. Classes that are engaging instead of boring; classes that are community based and not individual; classes that are worth their time and are relevant to their lives and goals. These vision of what CS education should be go against the flow of current perceptions and practices, akin to a boat going upriver of how students think we teach CS today. We visualized students' active dreaming in Figure 4.3 to highlight the effort required to design and build against current practices in CS

education. Furthermore, while we asked students about CS education, they continually expressed their interest in a more expansive computing education. Students were interested in learning about how CS could be integrated into other topics and subjects. They saw CS as a tool that should be integrated to help them learn other subjects and understand the world around them.

This work provides insights that corroborate what we previously know to be good pedagogy. What is novel is that the students were the ones to identify and voice a desire for CS classrooms with environments, practices, and characteristics that we know support effective learning. For example, much like prior work shows the importance of adapting materials to students' interest in the classroom (Hwang et al., 2023), these students voiced the desire for CS education to be *meaningful and relevant* to them. Scholars have also pointed to the importance of *autonomy* and *collaboration* (S. L. Johnson et al., 2025) which these students identified as essential to CS education, and the need for educators to have strong pedagogical content knowledge (Yadav & Berges, 2019) as a sign of their *accountability*. Additionally, students envisioned CS learning environments as warm and inviting spaces. As pointed out by Moreno, teachers work hard to create such enriching environments because they recognize the impact of these unseen labors on student learning (Moreno et al., 2024).

Beyond aligning with what scholarship points to as good pedagogy, students' visions of CS classrooms echoed tenants that are present in feminist pedagogies. Students want *choice and autonomy* in their learning, as constructionism foregrounds (Papert & Harel, 1991). They wanted to *do relevant work* as Culturally Responsive Pedagogy (Ladson-Billings, 1995), and Culturally Responsive Computing (Scott et al., 2015) call for. Finally, students felt it was important to grapple with and do *work that mattered* and had a broader impact on the world as Culturally Relevant Computing, and the Kapor Center Framework recommend (K. Davis et al., 2021; Gay, 2002; J. J. Ryoo & Blunt, 2024). Taken together, these pedagogies along with the care for others that students described in this work, are feminist pedagogies. Our work elucidates the importance of listening to students when designing CS education, especially given that this data shows that when we listen to students, they end up advocating for what we in prior research found to be effective pedagogies in the classroom. If we listen to students, and implement their suggestions,

we are able to both provide appropriate learning environments, and build CS education in a way that will keep our students engaged.

4.5.1 Limitations

Our methodological decisions introduce the possibility of threat to the validity of our findings. Although we worked to mitigate them, power dynamics between session organizers and students may exist in such a research setting. The positionalities of the organizers and researchers may also have impacted what students were willing to share in their discussions and reflections. Additionally, although we worked to bracket them, the positionality of us as researchers influenced how we understood and analyzed the data. Our work is just one study: the themes that we observed are likely closely connected to the specific organizers, students, and location. These students self-selected to be part of this voluntary extra-curricular group. They live in a city with many technology based companies and attend US public suburban schools. It is important to note because their opinions and perceptions may not represent all student's desires and needs. Additionally, these students only have their lived experiences, which all include attending school. It is possible that it could be difficult to imagine alternate educational realities which might not include school. It is crucial to study the beliefs and desires of student populations with backgrounds not represented in this work.

4.5.2 Implications

This work points to the urgency and importance of re-examining, disrupting, and re-imagining what CS classrooms, environments, and expectations should be so that they are not only teaching the content of CS, but also appealing to students. We need to work with youth to shape their education: although they are not yet experts on content, they are experts on their own experiences of learning. Building with teens might require more time, but it will also mean we are investing in classes and educational experiences that matter to them.

Future Research. In addition to seeking youth voices in other populations of youth (e.g. rural

students, young women, students who failed CS), future research should explore how to prepare teachers to teach CS so that all students are excited to enroll and find their courses worthwhile. We must also study the effectiveness of these pedagogies across student populations. Research should also pay close attention how students feel about the pedagogies we currently use in CS education, as they influence their experiences, motivations, and computing identities. The contributions of this work regarding the pedagogical characteristics which may enhance secondary students' experiences in the classroom may also be tied into existing efforts in CS education like the CAPE framework (Fletcher & Warner, 2021; McGill et al., 2023) by providing a more nuanced understanding of the experiences students want and need in a CS classroom. Further exploring the pedagogies secondary students want CS education to adopt to enhance their experiences may have implications on participation in CS spaces.

Practice. Concretely, we offer a few recommendations for researchers and educators as they are implementing secondary CS courses. First, make sure class content, projects, and environments are visible within the school. Students were unwilling to take the risk to enroll in the 'black box' that CS classes were, and because the cultural assumptions about CS are broadly negative, they are assuming those courses are math-y, boring, and lonely. Second, work with students to design projects that are relevant and interesting to them, and that allow students to bring their own experiences, expertise, and interests into their work. Third, extend coursework beyond just learning content to doing work that has a positive impact on the broader community.

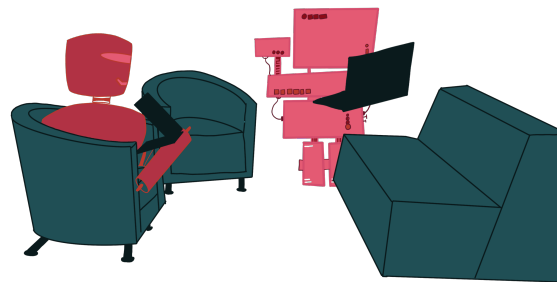
We are at a moment in time where we are rapidly building and offering new courses (Code.org, 2022) and this is a time to push back on the problematic assumptions and cultural norms held around CS. Because we built trust with students and were responsive to their desires and feedback throughout this work, students were able to be vulnerable and honest in sharing their thoughts about CS and in designing CS education environments, similar to the importance of trust in the work by Roldan et al. (Roldan et al., 2022). It is therefore important to be responsive to students and to build this trust in CS classrooms.

We know from Ryoo and Blunt's work that students view computing as a form of power, and they want their CS courses to be critical and address the social responsibility required of

computing (J. J. Ryoo & Blunt, 2024). To engage students in critical work, we need to build trust within classrooms so that students feel free to engage with critical work and topics (Everson et al., 2022). Building trust come from creating learning spaces and practices so that students feel they've been heard, and are excited to enroll in classes where educators are attuned to their needs to make impacts both within and outside the classroom. In order to build trust with students so that they can do critical work that matters, we need to listen to them, and acknowledge their desires when it comes to CS education. Therefore, it is essential to build environments where student voices are heard and valued.

Chapter 5

JOYFUL ENVIRONMENTS: STUDENTS' EXPERIENCES OF JOYFUL SECONDARY COMPUTING



This work was completed with analysis support from Rotem Landesman, Janet Jiang, F. Megumi Kivuva, Eman Sherif and Amy J. Ko and just accepted for the International Computing Education Research Conference in 2026.

5.1 What is joy in computing education?

For the next two studies, I wanted to ask students and teachers directly how they experienced joy in computing classrooms. This task was much more difficult than I anticipated it to be. Whenever I mention that I am studying joy in computing, faces light up and I get eager excited questions. I anticipated the same when asking students and teachers about their experiences.

However, when I started recruiting students to interview, I had a low response rates. In past research, like the rigor study in chapter 3, I've found more than enough participants very quickly with a single email to the undergrad CS undergrad mailing list. For this study I started by recruiting from the CS undergrad mailing list. I also emailed the iSchool undergrad mailing list, and then asked collaborators studying CS in the Pacific Northwest to distribute broadly to

their undergrads (5 additional universities probably totaling more than 3000 students). I got 15 responses, 10 that experienced joy in secondary CS. Only 5 responded to interview requests.

A few months later, I had to recruit a second time, and this time I extended my recruiting requests. I again emailed the previous UW mailing lists knowing that it was a new school year, and there were several hundred new students in the undergraduate programs and on the mailing lists, and maybe students had more time in early fall than they did 6 months earlier in the spring. I also asked a favor of all of my former labmates turned professors and any CS education friends in other universities, they shared the recruiting form to additional students in North Carolina, Maine, Wisconsin, Illinois, Colorado, and Indiana. I received 10 additional responses, 8 of whom indicated they experience a joyful secondary CS experience. Of the thousands of students emailed, only 18 people reported they experienced joy.

I also found it was really hard to get responses when recruiting teachers . I distributed 105 design probes (with 3 pre-stamped pre-addressed postcards) at CSTA and emailed 72 teachers 3 times during the school year asking for them to mail back a postcard or reply to a quick anonymous 2 question online form linked in the email. Of those 315 postcards, teachers mailed back 10 postcards months and digitally responded 17 times. This response rate did not match the enthusiasm they had when grabbing a design probe. I get it, teachers are busy, who is going to respond to an email from a stranger asking one more thing from you? Physical design probes are hard to find when you need to mail it back at a certain time. The world feels like it is crumbling around us and we ask so much of teachers. I get it. I've had to triage when I taught. I'd rather teachers invest those moments with a student than write me back.

This low response rate is in contrast to the excitement that I hear when I share that I'm studying joy in computing. People are hungry to hear about joy in CS classrooms, and yet it is hard to find people who think of CS classrooms as joyful places and who are able or willing to talk about it.

5.2 *What makes computing joyful for students?*

All classrooms, including CS classrooms, should be places that value humanity and center the learners' experience (hooks, 1994). These environments should foster a sense of belonging (Fletcher & Warner, 2021), as prior research makes clear that positive experiences encourage learners to keep learning (Jane Margolis et al., 2017; Margolis & Fisher, 2002). One word to describe belonging and the desire to continue learning is *joy* (hooks, 1994).

Scholars are already engaged in understanding joy in learning, giving talks about joy (Roque, 2024), and designing pedagogy for joy (Jasinski & Lewis, 2016). There has been some research studying joy in CS in particular. Sometimes instructors seek to design for their own joy (Wilkens, 2003). Others learned that not all formats of learning computing are joyful for all students (Parker & Hankins, 2002). Others designed curriculum intending to produce joy and found that it encouraged broader participation (Garcia et al., 2009; Harvey, 2012; Mark & Klein, 2019; Mönig, 2020). Some have designed for joy in computing for particular populations; for example, centering connection in elementary CS learning (Karlin et al., 2023).

Despite all the intentions to engender joy and what we have learned about joy, prior work has yet to ask secondary students themselves about their experiences of joy, or what joy is for them. This gap is notable: all educators know that their intentions to foster joy do not necessarily produce joy. Moreover, without an understanding of what students themselves say is important about fostering joyful CS learning, educators, teacher educators, curriculum designers and other key stakeholders in shaping CS learning have little reliable guidance to foster joy. And without that guidance, secondary students first exposures to computing may be their last, preventing them from wanting to continue to learn and explore computing concepts.

To understand secondary students' experiences with joy in CS education, we asked two questions:

- What do joyful CS learning environments have in common?
- What pedagogies support joyful CS learning spaces?

To answer these questions, we reached out to 10 universities' CS programs over two rounds of

recruiting, hoping to find many to interview about their secondary CS learning, along with their former teachers. To our surprise, we found very few students who reported joyful secondary CS experiences. But the small set of 8 that we did find, along with their teachers, have much to teach us about what pedagogies and values led to their joyful experiences.

What this range of work in CS education shows is that instructors are designing courses in the hopes that students find the same joy in computing that they themselves have. There is an assumption that what instructors find joyful, their students will also find joyful. While researchers have studied student enjoyment, there is a dearth of research examining joy in K-12 educational settings or from the student perspective.

5.3 Related work

Before I share the methods and findings from this chapter and the next, I need to share the prior work studying joyful pedagogies and joy in CS.

5.3.1 Joyful Pedagogy

Sometimes, prior work centers instructors' joy. In one instance, a teacher centered joy in his classroom by focusing on community, visibility, and agency, defining joy as "deserving of study and investigation as any other pursuit" and that joy is "part of the work." This teacher noticed students were chattering in Zoom chat and other technical interfaces, they worked with students to design a Discord channel for that purpose, students created a code of conduct and maintained the channel. (Crawford-Garrett & Carbajal, 2023). Another joyful pedagogy text argues for three factors: agency, community, and creativity, but again centers the teacher's joy (Camfield, 2025). These works suggest that it is sometimes essential that teachers and instructors experience joy in teaching, but this work is done from one instructor's perspective. This work also does not connect or examine students' experiences of joyful learning.

Other prior work in higher education has broadly examined the phenomena of joy. In one study, researchers found that the inclusion of culture and identity, multiple pathways for knowledge generation, acceptance and respect, collaboration, and meaningful approaches to teaching content

were ways to promote joy in learning (Omidire et al., 2021). Researchers also argued that joy can be a radical subversive pedagogical act of defiance, pushing against oppressive structures. Like humor, joy is relational and embodied (Hall & Steele, 2024). O'hara et al. argue that, "If we want to rekindle joy, we need to resist the impulses that drive us to conceptualize education as a transactional exchange for quantifiable outputs and economic worth." and that trust is the foundation of joy (O'Hara et al., 2025).

Some prior work has examined joy in specific disciplines. In writing education, Muhammad argues that the purpose of education must be to ascend, to build a fulfilling life through learning. Joy itself is a worthy educational pursuit, and students must be exposed to joy in reading so that they can live and write about it (Muhammad, 2022). In art, Jasinski and Lewis argue that there are many kinds of love, which lead to joy, but the ideal love is to love students no matter who they are or who they become (Jasinski & Lewis, 2016). Seeking to understand Joy in STEAM, Scipio et al. developed Pedagogies for Joy (POY), which attend to affective experiences of participants, cultivating third spaces by drawing on diverse experiences so all students are supported, leading learning, and dreaming of pathways through learning (Scipio et al., 2025). Responding to the POY, Ryoo calls for more work that focuses on the voices and perspectives of young people (J. J. Ryoo, 2025). In other work, Ryoo identified the importance of humor in the CS classroom and found that humor and the resulting joy were useful for student engagement, positively influenced students' views of CS, leading to engagement in new forms of thinking and learning." (J. J. Ryoo, 2019a). This work also found that humor also supported critical thinking and communicative competence. These works seek to center joy, but have not yet examined joy from the perspective of students, and has only offered us what a limited number of teachers perceive as joyful.

Joy in computing education

In CS education research, we see a variety of definitions of joy that are different from the definitions used in this dissertation, often conflating it with the word "enjoyment," where students are asked if they enjoyed a course, and then researchers write about the joy students experienced. While joy and enjoyment are both positive, they are different emotions and have different outcomes.

Often joy is declared because of instructor or curricular intent (Barnes, 2013; Garcia & Harvey, 2011; Garcia et al., 2009; Harvey, 2012; Mark & Klein, 2019). It's also been used to design new programming languages (Mönig, 2020), and to foster engagement in a first year engineering course (Meza et al., 2020). Sometimes instructors center their own joy in teaching (Parker & Hankins, 2002; Wilkens, 2003). Other instructors designed for what they perceived students should find joyful in online learning (Dijkstra, 1988; Woodman et al., 1996). Practitioners design for joy in particular populations, like elementary students (Ivey et al., 2021; Karlin et al., 2023; H. H. Lee et al., 2023).

Some research examines how important, and difficult it is to create spaces that center joy. Additionally, Jones and Melo call for more critical conversation in CS education by imagining alternative visions of CS that centers student joy and liberation (Jones & Melo, 2020). In a study examining the practices of creative coding instructors, McNutt et al. found that these practitioners were designing for “slowness (how friction can make room for reflection), politics (including the lasting effects of particular technologies), and joy (or the capacity for playful engagement).” instructors shared that certain amounts of competence were required to experience certain types of technical joy, and that all tools should have playful but not toyish elements to best support student work (McNutt et al., 2025). This call for joy and study of joy do center joy and we still need to hear from secondary CS students and teachers.

Additionally, prior work has also examined community as a tool to increase student retention and enjoyment of the subject of computing. For example a study on pair programming found that all students had increased confidence, and long-term persistence (McDowell et al., 2006). Additionally, Porter et al. included pair programming, media computation, and peer instruction and found increases in retention and enjoyment (Porter & Simon, 2013). Chavez shared the battle against course creep and deadlines improved outcomes in a data structures course (Chávez, 2023).

While instructors are working design for joy in post-secondary CS, the environments and requirements of K-12 classes are different. We need to understand what brings joy to K-12 instructors to help build joyful learning environments for their students and retain teachers in the classroom.

5.4 *Methods*

We conducted qualitative research using Braun and Clarke's 6 steps for Thematic Analysis (Braun & Clarke, 2021; Patton, 2015) in order to better understand what students' identified as joyful environments in secondary computing environments. To unearth these experiences, we interviewed participants who self-identified as having joyful secondary computing experiences (Braun & Clarke, 2021). We also asked student participants to identify teachers in their joyful experiences, and interviewed those teachers. This study was IRB reviewed and exempt at the researchers' home institution.

5.4.1 *Positionality and Reflexivity*

The research team approached this work with joy, care, and curiosity. We are a team trained in CS and education. Many of us carry identities of educators, and want to understand what makes classrooms joyful, CS and otherwise. A number of team members feel like outsiders looking in to CS, and brought that lens to this work. We bring our experiences and our positionalities to this work, and that influenced our interpretations. We balanced these personal perspectives with methods that explicitly focused on accurately capturing and valuing the voices in our the interviews.

5.4.2 *Participants*

As students' experiences were the guiding point of this study, we started by recruiting what we described in our materials as "students who had joyful secondary (high school) computing experiences." We left it to the students to define what a joyful experience was for them. We conducted 2 rounds of recruiting. In Spring 2025, we emailed undergraduate mailing lists in our home institution in computer science and associated fields, and asked our personal networks to share in their courses. We received 15 responses; ten of those met inclusion criteria, five responded to interview invites. In Fall 2025 we again emailed our undergraduate mailing lists, and expanded our asks of personal networks at other institutions. We received 10 additional responses, 8 who

met inclusion criteria, and 3 responded to interview invites. For students, our inclusion criteria was students who had joyful secondary computer science environments. Our exclusion criteria was students who had later post-secondary joyful computing environments, students who did not have joyful secondary computing experiences (and really wanted to share how awful they were). We think it is important to note that although our recruiting message was sent to thousands of students, only 18 met the inclusion criteria of a joyful secondary computing experience. This is likely reflective of how rare a joyful secondary computing experience is.

We then recruited teachers nominated by the 18 student respondents who met the inclusion criteria. From the first round of student recruiting, 5 teachers were identified. We reached out to all through LinkedIn or email. Two responded to our requests and were interviewed. From our second round of recruiting 3 were identified, and 3 responded to our requests and were interviewed. Our only inclusion criteria for teacher participants was that students identified them as the educators in their joyful computing environments. Participants and Student-Teacher dyads appear in Table 5.1.

5.4.3 Data Collection

The first author conducted semi-structured interviews over Zoom, each for a duration of 30-75 minutes, and participants were compensated \$25 for their time. The team conducted pilot interviews with community teachers and students until the research team was comfortable that the environment was identified. As noted previously, we left it to students to identify and define what a *Joyful CS Environment* was for them.

We asked students:

- 1) Tell me what the class was like,
- 2) Tell me about your favorite experience or project in CS,
- 3) Tell me what you enjoyed about your CS class,
- 4) Tell me what was challenging about CS,
- 5) What were your experiences with your teacher?
- 6) Were there any class policies that stood out and made that class great?

Joy Identifier	Joy Educator	Joy Environment
Student 1	(teacher never replied)	Outside of school, weekend CS course, multi-year
Student 2	Teacher A (identified during interview)	APCS, Advanced Topics TEALS classroom (Teacher A was a TEALS instructor)
Student 3	Teacher A	APCS, Advanced Topics TEALS classroom (Teacher A was a TEALS educator)
Student 4	(teacher never replied)	APCSP
Student 5	(teacher passed away)	Oceanic program for the last two years of secondary school.
Student 6	Teacher D	Specialized School: No specific CS courses taught, instead multiple instructors incorporated student's interest in CS into other coursework (Teacher D taught Math)
Student 7	Teacher C	APCS
Student 8	Teacher E	APCSA followed by an Independent Study. Student 8 found the independent study the joyful part they wanted to share.
	Teacher B (nominated but student never scheduled an interview)	APCSP and APCSA

Table 5.1: Table of the students, joyful educators, and brief context of the environment of joyful CS

- 7) What was grading like?
- 8) What was feedback like?
- 9) What was the class environment like?
- 10) How did you feel taking these classes?
- 11) Any information that I haven't asked about?

We asked teachers:

- 1) Tell me about your teaching journey,
- 2) Tell me about your teaching philosophy,
- 3) Tell me about a great experience you've had teaching CS,
- 4) What do you consider a classroom success?
- 5) How do you know a student is learning?
- 6) How do you approach a struggling student?
- 7) What environment are you trying to build in your classroom?
- 8) What classroom policies have you found helpful?
- 9) What was grading like?
- 10) What was feedback like?
- 11) Any information that I haven't asked about?

After we conducted the interviews, we emailed participants their compensation and a brief survey to collect any demographics they wanted to share and any additional relevant information that came to mind after the interview. All transcripts were verified by the research team as part of data familiarization (Braun & Clarke, 2021).

5.4.4 Analysis Process

The analysis process took place over four months. The research team met weekly and took memos of our thoughts individually and our collective conversations throughout the entire process. Additionally, each member of the research team spent time reflexively journaling about their own joyful computing environments and experiences of joy in computing. Using student transcripts, the team individually identified themes of environmental factors that were part of the student's joyful

Element	Theme	Definition
Relating to Students to Center the Student Experience: Laying the Groundwork for Trust	Relationship	Teachers established relationships with students that respected them as individuals. These relationships were foundational to mentor students through the learning process.
	Teacher Presence	Aspects of the teacher's presence that supported their ability to learn. Teachers utilized humor and care to create an environment where students were able to learn CS.
	Collaboration	Collaboration with peers and instructors helped with learning and allowed for more complex problem solving, also not competitive. Another aspect of collaboration was instructor humility, instructors would share that they were didn't know something and would need to learn with students.
Prioritizing a Relationship with Learning: Building Trust	Attitude about CS	Teachers made learning over work, and emphasized the importance of making mistakes in learning
	Assessment Considerations	Teachers decoupled learning from assessment and prioritized the learning process over grades and outcomes. Sometimes that meant assessments were low stakes and expectations were clear.
	Assistance to Support Learning	Teachers worked to provide support to students outside of class, or as individual students needed.
Framing CS with Purpose: Building Motivation for Students	CS as a tool	Teachers framed CS as a tool to learn with rather than an end goal, they incorporated student interests into the assignments and worked to make CS interesting enough that students were motivated.
	Meaningful work	Teachers worked to align projects to relevant and meaningful work for students by attempting to capture student interest and to motivate students. Additionally teachers worked to make learning CS fun.
	Responsiveness	Teachers were responsive to student's previous knowledge and experience, and worked to tailor assignments for individual students both in topic as well as with flexible deadlines and requirements for assignment submissions
Reification of a Joyful CS Experience: Trust Established	Embodiment	Teachers introduced complex topics with tangible ways of learning, making things students could see, and touch
	Projects	Work was often project-based which meant that student work was embodied because students were creating programs, or physical things. Additionally, projects allowed for students' autonomy in selecting topics and teacher responsiveness in designing so that students cared about the work they did. These projects often went hand-in-hand with flexible requirements and deadlines because projects varied between students.
	Autonomy & Agency Challenge	Students initiated their own learning, or made long term goals related to CS, sometimes making long-term goals related to CS. Many students mentioned that complex material was presented in a way that they could really understand it, and they found they had learned far beyond the expectations of a secondary CS course.

Table 5.2: **Codebook of Themes** and the associated elements in a joyful CS experience.

experiences. Then themes were identified in group discussion, before the team affinity diagrammed the themes. Next the team identified pedagogy that contributed to these environments of joy from the teachers' interviews and affinity diagrammed those factors to arrive at themes. Once both sets of themes were identified and reviewed we matched the themes between students and teachers and defined the codebook seen in Table 5.2. Initially, some of the themes that the students found essential to a joyful environment (*Collaboration, Responsive Teaching*) were not in the elements identified as themes from the teachers' interviews. However, when re-reviewing the teachers' interviews, we found evidence of both of these themes. We found that students identified 4 elements of a joyful CS experience. Each element had several components. Each student identified all four elements as part of their joyful CS experiences, but not all of the components of each element.

We resolved all disagreements through discussion. For example, researchers debated whether humor was its own code, or an aspect of a *Teacher Presence*, ultimately deciding to collapse it into *Teacher Presence*. Finally, we defined themes and then re-read each transcript, attending in particular to student-teacher dyads, refining codebook definitions.

5.5 Elements of Joyful Computing Experiences

We found that students identified 4 elements of a joyful CS experience: *Relating to students to center the student experience*, *Prioritizing a Relationship with Learning*, *Framing CS with purpose*, and *Reifying a joyful CS experience*. Figure 5.1 shows these 4 elements, aligned with growth in trust and student agency.

5.5.1 Relating to Students to Center Student Experience: Laying the Groundwork for Trust

Students who had joyful CS experiences noted that teachers valued relationships in the classroom by working to know and value students, and by establishing an environment where students could also build relationships and collaborate with each other. The three components of this element were relationship with students, teacher presence, and collaboration.

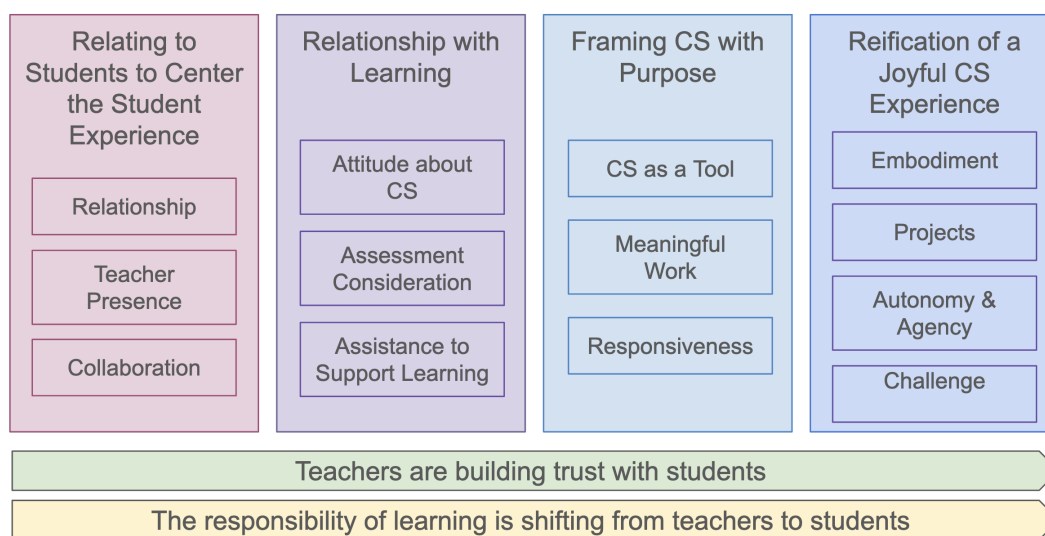


Figure 5.1: The research team conceptualized the reification of a joyful classroom with the above diagram of four elements of what Teachers were doing to create these joyful environments. Each of the four elements has several different components. Not all students mentioned each of the components, but they did mention more than one component from each of the four elements.

Relationship

Teachers established relationships with students that centered respect for students. These relationships were foundational as teachers mentored students through the learning process. This element was mentioned more often by students than teachers. For example, Student 4 shared how her teacher worked to make students feel like the whole class was part of a team to learn CS, "[H]e was, like, a very chill teacher... I think he did a great job of making us all feel, like, We're kind of, like, in this together." And student 8 mentioned how teacher E knew him well enough to encourage him to pursue CS after he completed an initial CS class because he thought, "the computer science teacher was awesome, and she prompted me to, like, go to [Local University] for early classes, and just encouraged me to keep learning." Student 8 instead opted to create an in-school independent study. He collaborated with his teacher and two peers to design a year-long course of work that allowed the students to learn C++ and the relevant frameworks and to build

a game. Teacher A shared how building relationships with students was one of the hardest things for him to do, but that he knew it was valuable to reciprocate whenever students would share anything about themselves:

" I will say that one of my weaknesses is that I find [building relationships] hard to do.... I'm quite an introvert. And so it's hard to go up to someone, and without it being too artificial, and without it being too deep, because that would be awkward, but like finding that middle ground where you could start a relationship, and then sort of have that grow. It was hard for me to be the initiator. Fortunately, students would find ways when topics were really interesting to them, or topics were really scary to them, and for whatever reason they just needed to talk about it. " – Teacher A

Later he mentioned the advantage of being one of the only CS teachers at his school; he would have some of the same students for multiple years, allowing trust to grow so he could effectively mentor students on independent projects in advanced CS classes.

" I was doing a bit of mentoring as they were working on their own project, and this only worked because it was a student I had developed some kind of relationship with over the past 2 years prior in class." – Teacher A

Students shared how important it was to be noticed and encouraged by their teachers. Student 4 shared that her teacher constantly encouraged her for her work, saying, *"He was very supportive and helpful, and very encouraging as well. He was 'y'all are doing great!' or he'd be like, '[Student 4], you're amazing,' and I'm like, 'Thank you!'"*

Teacher Presence

Many students shared aspects of the teacher's presence, like enthusiasm, care, and humor that supported their ability to learn. Student 2 shared how important it was to have an approachable teacher who was animated and excited to be there, almost as if he put on the mantle of his teacher presence every day:

" I was really scared and nervous, but [My teacher] made the field of computer science and programming feel really approachable, like, I think he's really gifted in explaining things clearly and succinctly to youth, like high schoolers. I remember just him being a very animated teacher. " – Student 2

Teacher C shared his focus, *"I teach the student, not the curriculum,"* and later shared how seriously he took bringing the appropriate presence to the classroom:

" I tell them consistency is always my big thing. They would have no idea... my house could have burned down, and I'd come in, I'd be the same thing. Like, it wouldn't matter. we got stuff to do, we can have fun doing it, let's leave everything else at the door. This is CS here, or math, and let's just kind of roll in." – Teacher C

Student 7 shared the impact that Teacher C's consistency and enthusiasm had on the classroom environment when she said, *"He was always, like, super energetic, and I feel like when he's energetic, like, it makes all the students feel, like, more energetic and uplifted, and they, like, kind of wanted to, like, mirror how he was."* She shared that he made class interesting and approachable. Even though she was initially terrified to take CS, she took a risk enrolling because she knew from a previous math class that she could trust Teacher C.

Students also mentioned how teachers used humor to alleviate the stress of learning CS, but also to build relationships and demonstrate that they knew the students. Student 2 shared that her teacher's humor helped to build a culture of ease.

" And yeah, not only that. He just cracked a lot of jokes in the classroom... And I think that's what made the class so fun as well. He kind of just treated us like we were peers, in a sense, and we were all just like friends getting along. " – Student 2

Teacher A mentioned that he would try to create assignments that mattered, but would highlight when energy in the class was low by making jokes like, *"How are you guys even still in this class? It is so boring."* He then used those jokes to incorporate the next necessary step in order to

motivate students. Teacher C shared he also used humor as a motivating tool to get stuff done when he said, *“we have a good balance of joking around and getting stuff done.”* Humor was a way he built common ground with students and created a lower stress environment and to build relationships with students, *“the kids laugh, and then they. . . Almost like they see we’re actual people.”*

Many students shared the passion their teachers brought with them to teaching. Student 3 shared his teacher’s dedication to the craft of teaching, *“[T]here was an element that he lived. . . he lived to teach”*, when sharing the lengths that his teacher would go to to make sure every student had the support they needed. Student 6 shared how their teachers knew they liked CS, and because they also enjoyed CS worked to incorporate it into their math and science coursework:

“ They had, like, a separate interest in computer science that they really wanted to share with people, I think... And they almost were able to, like, share their hobbies, because I know they were. . . like, ‘oh, if you’re interested in this, you should talk to me about it.’” – Student 6

Student 5 learned CS by analyzing data she collected on a boat as part of her hands-on classroom. In this secondary program, students each had individual research projects and used the statistic programming language R to analyze data and specimens to answer their own research questions. Student 5 shared that the teacher was so excited to show them the power of CS:

“ When she was teaching R, it didn’t even feel like. . . she was necessarily an expert. . . It was more like, ‘hey guys, like, look at this super cool thing that I know how to do...don’t you guys want to try this out too?’. And [seeing] that enthusiasm, I think, also made coding feel a lot more approachable... it was almost like a puzzle. ” – Student 5

Through teacher presence, humor and enthusiasm, it was clear to students that CS was a joyful tool, and learning it was a worthwhile endeavor.

Collaboration

Collaboration with peers and instructors supported student learning and allowed for more complex problem-solving. Students also compared these joyful experiences in CS to other competitive CS experiences they had. Several students shared that they collaborated with peers to solve problems they could not solve on their own, and developed relationships and friendships that endured beyond that class. Student 4 shared this collaboration and camaraderie contributed to the joyful learning environment:

“ [I]t was a really, like, friendly and good environment...I didn't feel intimidated by other people in the class... These are my people, these are my peers. that also probably helped contribute to, like, 'Oh, yeah, I like that class.'” – Student 4

Student 8, who took an independent study with two peers, shared how they would all work on a complicated problem when they got stuck:

“ I remember there were a couple days where I just spent the full hour looking into one problem, one issue, and trying to research it and learn it. And because there were three of us, when one of us came to an error, and if they eventually managed to solve it... and then we'd all learn from it” – Student 8

Because these three students were working together, they had each other to rely on when they got stuck. Student 8 and his peers also collaborated with Teacher E to develop their coursework because although she might not have the CS concepts they needed to develop their game, she did have a strong education background and was able to support that work. Instructors were an important part of the collaborative atmosphere. They would often collaborate with students to find answers. Teachers were vulnerable, willing to admit what they didn't know, and problems solve with students. For example, Student 8's independent study meant they needed some advanced physics simulations they would ask Teacher E for support. And Teacher E wasn't afraid to admit she didn't know and answer and would out to local university contacts when students got really stuck. Student 8 shared, *“But sometimes when we got really stuck...there was one*

time where she actually connected us with some people at [Local University]." Teacher A, who was a full-time software engineer in addition to teaching CS, was not afraid to admit when he did not know an answer, and would problem solve with students and model for student how to break down and start a problem telling students, *"I have no idea, either, but we'll look it up and try to figure stuff. We can work on it together."* Several students mentioned that they avoided later formal CS experiences because they were perceived as competitive, and competition wasn't something they were interested in. Rather, these students pursued alternative majors like informatics to be able to continue their joyful CS relationship.

5.5.2 Prioritizing Learning: Building Trust

Learning was the ultimate goal, any assessment was secondary. This included decoupling assessments from learning, creating low stakes environments, encouraging students to make mistakes, and the time teachers devoted to supporting struggling students.

Attitude about CS

Teachers made learning more important than work, and de-emphasized mistakes. Student 4 shared they, *"felt accomplished because I was able to, like, do the assignments, and I felt like I was learning stuff from that class."* And Teachers B and C shared that they would not cold call students, but they would offer reward tokens like a small piece of candy to encourage students to take risks and normalize mistakes. Teacher B shared, *"You can make a mistake, and still earn a prize right?"*

Student 8 shared surprises that arose during his independent study. For example, the students were given locked-down school laptops that did not have enough storage allocated to their logins to save the files they needed in order to make their game. They had to find workarounds. In these workarounds, they surfaced several weak points in the schools security infrastructure. They shared these weak points with Teacher E, who would pass this information on to the school's IT department. Teacher E shared her perspective and the trust she had in these students:

“ I don’t even really know all the things that they did, but they found their way around things and I wasn’t overly concerned, because they’re such good kids, right? They weren’t gonna do anything with that information, but they were like, ‘[Teacher E], we were able to access all of this stuff,’ I think people realized ‘Oh, we gotta secure these systems’...[The students] were just honest with me about what they discovered. And I would share that with people discreetly, ‘Hey, maybe we should fix our systems a little bit.’” – Teacher E

Teacher E worked with the school to navigate what could have been a very difficult scenario. In this scenario, you can also see the trust these students had in Teacher E. They passed on sensitive information that would improve everyone’s security.

Assessment Considerations

Teachers decoupled learning from assessment and prioritized the learning process over grades and outcomes. Learning was decoupled from assessment in several ways; grades were de-prioritized and learning was emphasized, sometimes students had multiple opportunities to take or re-take assessments. Teacher B shared that she decoupled learning from assessments by *“having a retake policy where every test you do, you can retake up to 100 full credit, I think has really helped a lot in terms of fostering that mistakes are welcome.”* She also shared that she extended this to projects, and encouraged students to learn what they misunderstood and, *“fix up your responses so that you know how to do it, and then I’ll regrade it up to 100.”* Additionally, Teacher D shared that their school renamed grades so that students would not worry so much about achieving A’s and focus on pursuing their interests and learning. And Student 8 shared that because of the work they were doing:

“ The grades were just the computer science teacher, she saw us working, and saw us talking, and was like, ‘They’re learning stuff. Cool.’” – Student 8

And Teacher E shared that it was not difficult to grade Student 8 and his peers because they were constantly running into difficulties and having to work through problems.

“ Probably they thought it would be very linear. But it was so messy, right? It was just so messy..that’s how I knew they were learning, because they knew that they got stuck, right? And then knew enough about what they didn’t know to then go seek out the next thing and just kind of bounce around and piece it together.” – Teacher E

For Teacher E, the process of learning how to get stuck and then figuring out what to do next was more important than getting it right the first time.

In a more traditional CS course, Student 7 shared that her CS tests were, *“a lot less stressful than a lot of other tests”* because there were clear expectations and if she had done the work, she felt that, *“most of the time had a pretty good understanding of what was on it.”* Her teacher, Teacher C worked equally hard to make sure both rubrics and grading requirements were clear for projects:

“ Before I even grade it, I’ll say, ‘okay, you got the rubric, I’m gonna read off the rubric, you point to your code where you’ve got it.’ Like, if I said, ‘you gotta make an enhanced for loop.so I’ll say, do you have an enhanced for loop?’ And they’ll scroll through the code, like, right there. ‘Okay. Do you have this? Do you have this?’ So, And then if they don’t, then we can kind of talk specifically on that” – Teacher C

This process not only made grading low stakes, it allowed for conversation and ways to clear up any confusion about expectations or content.

Flexibility of assignment requirements and deadlines was also important for students. Sometimes life just happened, and sometimes, because students were defining their own assignments, it took extra time to complete a task. Student 5 shared that since their classmates had similar deliverables, but very different data and research questions, grading was flexible, and different for each student:

“ I think... we would have similar kinds of deliverables of, like, submit some exploratory data analysis; submit a few figures, submit your citations. But, I would say the way they were also graded was pretty flexible It’s all pretty customizable.” – Student 5

And Teacher B shared the importance of extending grace with assignments and deadlines because *“life happens”* and shared *“I’m not super strict on stuff, right? Like I make mistakes, you know. I would hope somebody would have the grace to give me a second chance.”* She indicated the trust she had in students allowed her to encourage them to ask for what they needed, and be honest about what they needed.

Assistance to Support Students

Teachers worked to provide support to students outside of class, or as individual students needed. Each student shared examples of how their teachers made time to support them and answer questions they had. Sometimes that meant a tutorial period, sometimes that meant during lunch, and in one case, Teacher A created a ticket system, and would get back to students in about 5 minutes. Student 3 shared:

“ I don’t know if if [Teacher A] told you this, but he would literally answer you anytime before 7 pm. Which is, when he would go to bed. You would get a response in like 5 min from him. But any question.” – Student 3

Teacher A mentioned this system, but focused on the way he that he sought out students that were struggling and met them an hour before school started (the school’s designated tutorial hour) for extra support and encouragement. Students described the patience with which teachers would answer their questions. For example, Student 7 shared her struggle with selection statements:

“ So that was something I always had, like, a hard time with. at first, so then I had to ask [Teacher C] to explain it a lot with me...I feel like that really helped me understand, and then over time...the struggle just got less and less until I understood”
– Student 7

Student 1 shared that, *“anytime we had a question or like we wanted feedback, we can just ask her, and she’d give us feedback.”* and Student 5 shared similar teacher availability, *“you could,*

like, reach out to her, make a personal appointment, whatever.” Over and over, teachers shared stories of tracking down struggling students, and students shared stories of how easy it was to find support.

5.5.3 *Framing CS with Purpose: Building motivation for students*

Teachers sought buy-in from students in a number of overlapping ways. They framed CS as a tool to solve problems with and they worked to provide relevant, interesting and meaningful work.

CS as a Tool

Teachers framed CS as a tool to learn with rather than an end goal, and they incorporated student interests into the assignments to make CS interesting. The end goal was not to learn CS, the end goal was to learn how to do cool things and how to solve problems. For example, Student 3 shared how Teacher A presented programming as a way to solve problems, rather than a discipline itself:

“ [Teacher A] never really told us . . . ‘Oh, you should be CS majors,’ right? Or you. ‘Oh, you should pursue programming.’ But he would give us these tools. ” – Student

3

And Student 1 shared, *“A lot of times we were using programming to solve really interesting questions.”* In Student 5’s coursework, they learned programming in order to conduct their own scientific research. And for Student 6, their teachers would allow them to complete work as they needed as long as they shared their process.

Meaningful Work: Interest and motivation

In these joyful environments, teachers worked to align projects to relevant and meaningful work for students by attempting to capture student interest and to motivate students. Student 7, who enrolled in the CS course because of the relationship she had with Teacher C from a previous

math course, shared how her hesitance to code faded away when she realized all the things she could do. She stopped even realizing she was doing the math and programming because she was able to make her programs interactive. She was both interested and motivated to do work she found meaningful.

"The designing aspect of it was definitely my favorite... seeing the code turn into something visual and interactive. That was my favorite portion, because I was able to put in all this work creating something. " – Student 7

Teacher E found another way to introduce meaningful work. She realized that her students needed a break from their year-long independent study project and introduced another opportunity to use the students' coding skills:

"We had sub-shortages, you know, teachers are covering each other's classes... It didn't seem like a super fair way for teachers to get assigned to do class coverage...I sat down with [the students and said] 'I've got a problem that we need help solving... I want a way to track, like, who's gone, who's next on the list, who's up, who's available during what prep hour... How can we make it fair and transparent?'" – Teacher E

She went on to share how they quickly figured out a way to program, automate, organize, and fairly distribute the sub list, so that the first teachers on the list were not constantly covering for everyone. The teachers felt relief that the system was established which allowed them to predict where they were in the sub list, so much so that they are still using the program the students had written several years later. Teacher B worked to interest and motivate her students by providing a variety of options to practice their abilities.

"So I had different versions of assignments that they could do so like I gave them a little menu... if you didn't want to do the practice like you could be a TA for me and help other students...Or you could do these kind of bonus extension projects. And I had some more challenge-type projects " – Teacher B

Student 3 shared a story about how much he enjoyed a project where he combined Conway's Game of Life with the classic game Space Invaders:

"[I]t was fun. And I think, most importantly, it was, you know, taking the project and going a step further, like taking that framework that you're given and making something new out of it. Which, honestly, I think creativity is one of the hardest things to encourage in a CS course" – Student 3

Being given the choice to define his own project, as well as incorporating elements of creativity, meant that a complicated task became fun for him. Student 3 was taught by Teacher A, who worked fastidiously to motivate students by offering choice and by making projects relevant for students. He also ensured the underlying complicated mechanisms were clearly explicated. Teacher A shared that he felt it important to explain why and how what they were doing was useful, rather than just doing it for the sake of doing it, because it was important to motivate students and make the work interesting. This also meant that in addition to enjoying what they were learning, students had a deep understanding of complex material (which will be discussed in Section 5.5.4).

Responsiveness

Teachers were responsive and worked to incorporate student's previous knowledge and experience and pop culture references. They also worked to tailor assignments for individual students both in topic, and through deadlines and requirements.

"[Teacher C] always did make, like, references to, like, things that were current... when he said them, everybody thought it was, like, so funny." – Student 7

And Teacher C shared that he did this because he knew that he had to be approachable and develop a strong relationship with students because he was one of the few teachers in the district that students could seek support from:

“ Being one of two teachers here that teaches it...;They kind of know how fast I can find or talk with them. They just know I’m not intimidating.” – Teacher C

Student 6 shared how Teacher D differentiated instruction and tied Student 6’s interest in CS into math class.

“He had a lot of computer science training... He would figure out ways to, kind of, to use my interest in computer science to support me doing other things. ” – Student 6

And Teacher D shared the flexibility he had to tie curriculum to student interests, *“Yeah, you could, you could definitely, kind of go off on, on sort of, like. Cater to your interests, the students’ interests. Try to find some overlap and stuff like that.”* Within this particular school, teachers had the ability to share their own expertise to student interests and curiosities. This autonomy for teachers and students will be discussed further in section 5.5.4.

5.5.4 Reification of a Joyful CS Experience: Trust Established

Students described several aspects of learning that established their CS experiences as joyful.

Embodiment

Teachers introduced complex topics with tangible ways of learning, building demos that students could see, and touch. They then created environments where students could also create things they could see, feel, and touch. Student 2 shared several physical lessons the teacher used, like the towers of Hanoi to teach recursion and learning types:

“He was teaching about object versus primitive types or something, and he had these 3D printed boxes with labels, like variable name labels. And then he was showing how like oh, with the primitive variable you could put a sticky note in this box. But if it was an object, it would be like this sticky note with an address that pointed somewhere else in the classroom.” – Student 2

And Student 3 shared the way that Teacher A also made every concept visible in order to help students understand abstract concepts:

“The best part about the class was that he did teach us the abstract concept, the algorithm. But you know, he linked it to something that wasn’t necessarily like real, but felt more real, something I could see and visualize and stuff like that. There was sort of like a custom library associated with every project that would render your output.” – Student 3

Additionally, Teacher C mentioned drones, Chromebooks, and the disassembly and reassembly of his personal computer as a way to make sure students understood the everyday technology they interacted with in order to build an understanding of the underlying components that made the programming visible.

Projects

In students’ joyful experiences, work was often project-based, which meant that student work was embodied because students were creating programs, or physical things. Additionally, projects allowed for students’ autonomy in selecting topics and teachers’ responsiveness in designing the projects so that students cared about the work they did. These projects often went hand-in-hand with flexible requirements and deadlines because projects varied between students. Several teachers articulated that these were not projects where everyone completed the same steps, but projects where students had choice and worked to define the goals and requirements. Student 2 shared that most of her coursework was projects and that those projects were *“super interesting.”* She then shared about several examples, including the first project that class did which was a *“Java program where you clicked keys on your keyboard, and it generated some like Midi sound, and you could make like a musical player.”* She clearly remembered the projects she completed years later. When asked about these projects, Teacher A shared that he initially inherited some projects but worked to make sure that they were fresh every year, and completely revamped several each year to make sure these projects were interesting and offered students autonomy and choice.

Student 4 often compared her joyful high school CS experience to a less joyful undergraduate experience. She mentioned learning just to pass tests in undergrad, and then completely forgetting the material. She compared it to learning CS concepts in secondary school that she felt she still remembered and used. She attributed some of this to the projects she completed in secondary class, and the flexibility that came along with them: *"You build a project, And you have, like, extra time to do that, and I was like, 'Oh, this is kind of cool.'"*

Autonomy and Agency

Students shared how they initiated their own learning and were encouraged to take agency over their work in these joyful environments. Sometimes, students took this agency to make short or long-term goals related to CS. They also often noted how important it was for teachers to also have autonomy to run their classes, and not be confined by unhelpful rules.

Student 8's entire experience was one of autonomy. He and two peers initiated a year-long-independent study to complete the giant project of building an entire game. He shared that the autonomy was, *"really fun, because I was just with two of my friends, and we were learning this thing... that all of us really liked learning."* He also shared that they learned a ton about project management, even though that was never one of their goals. They learned to better collaborate, and how to advocate for themselves. And he shared, *"It was definitely challenging, because we didn't have that structure. We were learning a lot, but it wasn't necessarily, like... Always useful information, you know?"* Learning wasn't linear for Student 8. He shared that his team had to go back to the drawing board and reconfigure earlier designed structures of the game because of new things they were learning. He shared that it was at times frustrating, but rewarding to have the freedom to learn this way. Similarly, Student 3 shared a story of a year-long project based advanced CS course where he could choose his own semester-long project and that he would be graded on the presentation at the end of the semester.

"And so at the time I was, I was pretty into chess. And so, as many programmers do, I was like, oh, I'm gonna make a chess engine...It was essentially a state space

search, which the only reason I sort of knew what that was was because [of an earlier project-base assignment.]” – Student 3

Due to that previous project and the complex topics covered, he was able to understand and make progress on that chess engine, and without the ability to learn difficult topics through projects, he “*wouldn’t have been able to make it probably.*”

But these advanced independent studies were not the only stories we heard of student agency and autonomy. Teacher 2 saw autonomy as essential and explicitly said “*I actually honestly think that giving students that autonomy and, you know, like having high expectations of them to like, learn how to manage their time.*” Students in traditional classrooms shared stories of starting clubs at their schools, or having the freedom to choose the focus of a week-long project. Student 6 shared that in science class they “*wrote a program to [transcribe DNA], because I was feeling very lazy, and they accepted that as the kind of thing that we could do.*” These small moments of autonomy were important to students and conveyed freedom and trust their teachers had in them.

In addition to the independent studies and projects students initiated, several students shared that they made long-term goals related to CS because of these joyful experiences. Student 4 shared that in the midst of her class she thought, “*wow, this is really cool, like, I think I want to do this in college.*” The goal of these experiences was not to push students to become Computer Scientists or to pursue CS, but several students shared that they chose CS as a career because of the power they saw in CS. In addition to several non-traditional environments, many students in traditional classrooms mentioned how teachers needed flexibility and autonomy to create environments students found joyful. Student 5 compared her current undergraduate course inflexibility to the flexibility and responsibility she was trusted with in her secondary program:

“I think they were pretty good about, you know, kind of tailoring the deadlines and deliverable expectations to what you needed and what you were expecting to do, right?” – Student 5

And Teacher D shared how helpful it was to have a school that trusted him with the curriculum and learning of students. *"One of the things about the school is that they really did allow, sort of, a lot of leeway in, sort of, the curriculum that you taught."*

Complex topics/Challenge

Many students mentioned in retrospect that complex material was presented in a way that they could really understand it, and they found they had learned far beyond the requirements of a secondary CS experience, even though these experiences were joyful. Rather, learning complex topics seemed to be a byproduct of a joyful experience. Students had autonomy to follow their curiosity. They would tackle really difficult projects in order to do meaningful work.

Student 1 shared that Teacher A *"made these really advanced like and complex topics still digestible for high school students."* And that, *"when I came into college, I didn't learn like these algorithms to like the end of my sophomore year,"* as a CS major. Student 3 also shared that he felt *"very privileged"* because of his secondary course and that because of this course he recognized how problem sets were not interesting because *"if you try to take the creativity out of something, you lose the problem solving."* and he was able to solve really complex problems because he had learned the art and creativity of problem solving in his secondary experience. Student 5 shared that her 3- term project collecting and analyzing a large amount of data was *"decently high level for a high schooler."* and that when she got really stuck she never felt overwhelmed because she was able to view it as *"a new puzzle, it's a new thing to, like, try and figure out."*

5.6 Discussion

We identified four elements of joyful computing environments that students perceived their teachers enacted in CS classrooms. These four elements show the relationality between processes of learning, teachers, and students. As a research team, we conceptualized the results as a "joy sandwich" of sorts. There were four overarching elements that each student identified, but the components of each element were not always the same. Each person may have their own sandwich preferences, much like environments that engender joyful experiences will vary from student

to student.

The elements we discussed above are building blocks that can be remixed to create joyful learning environments. These elements highlight two types of relationships that engendered joy: 1) joy emerging through relationships in the classroom, and 2) joy emerging through a relationship with CS. These two types of relationships have many ties to prior work.

5.6.1 Joy emerging through relationships in the classroom

Joy emerged through relationships in the classroom in several ways: between students and teachers, between students and peers, and in the way teachers helped facilitate students' relationships with learning.

Relationships between teachers, students, and peers. Prior work has conceptualized joy in rich ways (Roque, 2024). This work included a list of decisions facilitators make in those spaces, but has yet to examine the interactions between those decisions and student joy as this work does. In our work, we saw teachers moving with kindness, care, humor, and fostering a collaborative environment in order to *relate to students, thereby centering the student experience*. This aligns with previous work by Ryoo highlighting the importance of humor in CS classrooms (J. J. Ryoo, 2013). These environments built by these relationships nurtured collaboration over competition—in a way that allowed students to engage vulnerably; by making mistakes, taking risks, and by taking agency over their work. Centering *responsiveness* to understand student interests and experiences are key elements of Culturally Responsive and Sustaining Pedagogy (CRSP) (K. Davis et al., 2021). The fact that these elements surfaced strengthens CRSP's claims, and emphasized the need for these pedagogies in CS classrooms.

Relationships between students and learning. By building relationships with students, teachers were able to tailor course content to individuals fostering student relationships with learning. We heard students talk about how teachers who created joyful computing classrooms valued learning over work, decoupled assessments from learning, and made great efforts to assist and support students, which *prioritized a relationship with learning*. Much of this aligns with prior research we know about student's needs around assessments (Sherif et al., 2024), that students

need flexibility and care from instructors in order to feel seen in the learning process.

5.6.2 Joy emerging through a relationship with CS

Joy also emerged in the classroom when teachers framed CS, often through their own joy, so that students were able to build a joyful relationship with CS. With this relationship, students often continued their relationship with CS beyond the classroom, through hobbies or career choices. *Framing of CS.* When teachers were encouraging students to do meaningful work, and responding to student interests, knowledge and needs they *framed CS with purpose* and as a tool to build with. For example, when Teacher C framed a class assignment as *meaningful work*, Student 7 realized that she *"was able to put in all this work [and] creating something"*, and could now do things like design and make cute objects. In contrast, STEM tools often come with preconceptions about what you can and cannot do with them. Many of those preconceptions are gendered (Peppler & Thompson, 2024) and prevent students from imagining the world that is open to them. When CS was reframed, students saw opportunities for creation they had not envisioned, and those acts of engaging with CS were joyful. Alternate CS experiences were entangled with joyful CS experiences. It is therefore important to create joyful CS environments which provide alternate CS experiences for students to expand their conceptions of what CS affords, and what they can do with it.

Joyful relationship with CS. In our work, we saw teachers offer projects that incorporated student interests, autonomy, and agency as embodied work that made learning tangible. By doing so, teachers were helping to *reify a joyful CS experience*. Additionally, because students incorporated their own interests into their learning, they often motivated to learn complex content to achieve their goals. Students mentioned over and over again how CS allowed them be creative and solve problems. For example, Student 1 shared that, *"A lot of times we were using programming to solve really interesting questions,"* highlighting the framing of CS as a tool to problem solve motivated them to learn.

Throughout each of the 4 elements, trust played an important role. Teachers worked to build relationships and trust with students, and students trusted teachers to provide support to learn

CS, which aligns with prior work that showed students' desires to be trusted in the CS classroom (Everson et al., 2025). Because students trusted teachers, they were able to reify their relationship with CS, like Student 7, who enrolled in her CS course and trusted that Teacher C would support her. That initial trust made that student realize the doors to design that CS opened.

The goal of this study was not to understand student identity or interest development, although several students shared that they were already engaged in long-term goals involving CS and CS identities. However, we believe this study is orthogonal to the ideas of interest development (Renninger & Hidi, 2019), identity development (Lewis et al., 2016), and self-efficacy (Batra & Atiq, 2025). Instead, in our exploration of joyful CS learning environments, we saw the significance of *joyful relationships with CS* that students developed. These relationships with CS occurred because of students' strong relationships with teachers. These teachers were modeling the power and joy they themselves found as they related to CS. Student 5, for example, shared how her teacher enthusiastically embodied how cool it was to analyze data and see what you could learn using CS. Student 3 discussed how passionate Teacher A was, how he used CS to teach interesting lessons, and built tools to quickly reply to students. Both of these students built their own joyful relationships with CS by following the model of their instructors' relationships: Student 5 used CS to better understand her own data, and Student 3 shared the joy and creativity he found in the power of problem solving with CS.

5.6.3 *Pedagogy that supports joyful learning*

The findings from this study foreground the importance of relationships and highlight what is missing and often dismissed in CS education. Traditionally in CS, the quality and quantity of content is emphasized as most important to encourage learning. This study, however, finds that relationships are central to learning, and that content and deep learning follows. The type of pedagogies that we saw teachers in this work use to foster joyful CS environments were ones that centered care, relationships, students, embodied experiences (hooks, 1994; Tamboukou, 2018), value student agency, and autonomy (Freire, 2017; hooks, 2015; Woglom & Jones, 2024). Bringing these fundamentally feminist pedagogical approaches into the CS classroom offers educators

the opportunity to create joyful CS experiences for students, and are important to practice if we want students to continue to build joyful relationships with CS. To do this, the discipline must value relationships over efficiency and care over content. Our results suggest that if we fail to value the experiences of the students in our classrooms, our worries about content are moot.

5.6.4 Future Work

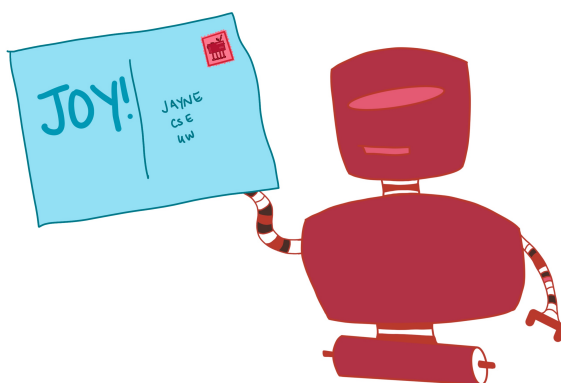
There are several limitations with this work. Many of the students interviewed for this study did go on to pursue CS or affiliated fields, this may be because our recruiting started with CS students and near peers. Results were also limited by the cultural context of the United States, and the pool of students who identified their experiences as joyful. Though we worked hard to faithfully to represent students' perspectives, the experiences and identities of the research team as well as our positionalities are inextricable. Another research team may have noticed additional or alternate elements and components of a joyful experience.

Our work shows that relationships underpin student's joyful experiences: teachers' relationships with students, the teachers' relationships to the discipline that they model for students, and the peer relationships that teachers' practices support. This suggests that it is fundamentally important that teachers have the time and space to value, nurture, and support those relationships in their classrooms. Our work also demonstrates that we need far more research on how to reliably create joyful experiences of CS learning.

More broadly, our findings suggest that future work is needed to explore feminist pedagogies in CS, designing CS courses with these student centered approaches. Future work should also examine joyful experiences in other age bands like post-secondary CS environments, and elementary CS environments. More work is needed to understand experiences of students who enrolled in CS classes, but chose identities and careers outside of CS. Additionally, there is the opportunity once researchers have identified joy in a field that is so often perceived as dry and boring (see chapter 4), to compare this joy against pedagogy in other disciplines, and perhaps make schools more joyful overall.

Chapter 6

TEACHERS' EXPERIENCES OF JOYFUL LEARNING: POSTCARDS FROM THE FIELD



Rotem Landesman and Amy J. Ko provided analysis support for this work. This work will submitted for review summer 2026.

6.1 What is joyful for teachers?

I love teaching, it is also incredibly demanding. I've never been as tired as when I was teaching. My roommate (who also taught) and I joked that all we could do at the end of the week was collapse, and we built rituals around that recovery. On Fridays we would have cereal for dinner, watch a show together, and often be asleep by eight pm.

My days were packed. I found I didn't work well if I took preparations or grading home with me, so I arrived early, and stayed late most days to finish that work. During the day I was often working with students and rarely had downtime.

I ultimately left teaching to go to grad school because while I was teaching, I didn't have the time to think about the cool things I was seeing students do in the classroom. I wanted to

better understand what I was witnessing, and the magic of learning. The joy of learning alongside students kept me in the classroom for a long time.

Other teachers leave the classroom for all sorts of reasons. We are losing teachers at a very high rate of 8 percent each year (Perez et al., 2025). They are leaving due to isolation, burnout, lack of support, and school policy (Perez et al., 2025).

One way to keep teachers in the classroom is to trust them to be experts and to do their jobs (Flagg Young, 1900). Which would allow them to have joyful CS classrooms, and to make the job of teaching one that is not centered on burnout and doing more, but is centered on the joy of learning.

No one has asked K-12 CS instructors what their perceptions of a joyful CS classroom are, and what it means to learn computing joyfully. In this study, we sought to do that by asking:

- What do secondary CS teachers think makes CS learning joyful?
- How do teachers conceptualize joy in a CS classroom?

To answer these questions, I asked a workshop of teachers to design a joyful computing environment, additionally I sent out design probes with CS teachers across the United States asking them to share their experiences of joyful computing over the school year.

6.2 Methods

For this study, we conducted a workshop of CS teachers in conversation at an annual conference for K-12 CS teachers and distributed design probes to CS teachers before thematically analyzing the data. We selected these forms of data because they offer two perspectives from teachers. The workshop in conversation over the summer, the design probes in reflection during the school year while they were teaching.

6.2.1 Data Collection

In order to triangulate, we collected multiple forms of qualitative data for this study.

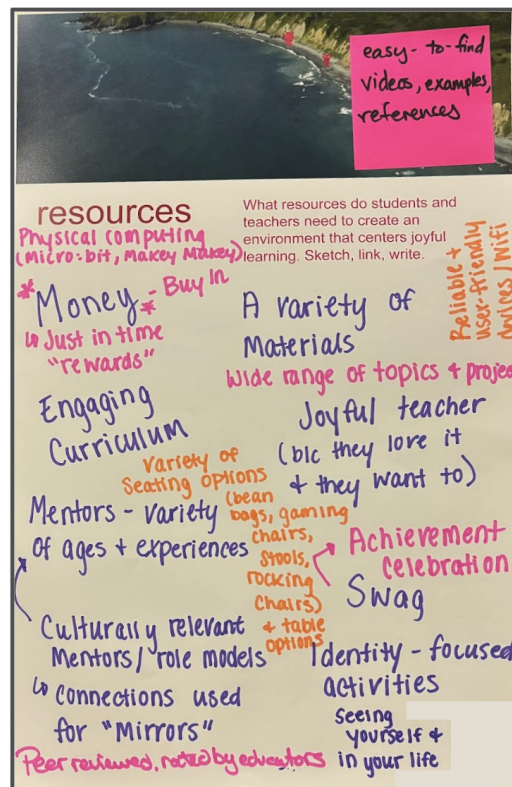


Figure 6.1: Poster from the resources small group written and collected at the workshop.

Workshop

In a one hour session titled "Dreaming of Joyful CS Classrooms" at a national K-12 CS teacher conference, we tasked teachers to engage in speculative design, a method where participants dream of futures that are preferable in all that is possible (Dunne, 2013; Khan et al., 2021). To do this, teachers broke up into five small groups and designed for architecture, pedagogy, curriculum, resources, as well as an open category. In these groups, the discussed the environment that created a joyful CS classrooms on a new island. Attendees had sticky notes and topic posters on each table to write on. Figure 6.1 shows an example of a completed poster. Once teachers had completed their posters, the session had a large group discussion to share what they had discussed. The first author recorded this conversation, after gaining consent from all attendees.

a Google form so that teachers would have a digital response option and also provided a space for "Is there anything else you want [the researcher] to know?" The Google form was anonymous. In total, participants mailed back 4 postcards and provided 3 digital responses for the first prompt, 3 postcards and 4 digital responses for the second prompt, and 3 postcards and 5 digital responses for the final prompt.

For the final prompt the research team also asked participants if they would like a cup of coffee "on us" for their time. Five teachers provided their emails, and received a \$10 electronic gift card.

6.2.2 Participants

In the workshop, and when handing out the design probes, we asked if participants were teaching CS, and used that as inclusion criteria for this study. As it was a conference for K-12 CS teachers, most people did. Several people declined to accept design probes because they did not identify as CS teachers.

6.2.3 Analysis

Guided by Braun and Clarke's process for thematic analysis (Clarke & Braun, 2017; Padiyath & Nelson-Fromm, 2026) we first familiarized ourselves with all the sources of data reading and listening to it multiple times both in raw format and then in spreadsheets. We each kept reflexive journals to capture our thoughts and experiences with joyful computing and our thoughts about the data throughout the entire analysis process. We then came together in analysis meetings to make a list of things we observed through our familiarization process. We created an initial list of themes we noticed in the data. We then coded all data with these initial themes before meeting again. We again examined all data and collapsed overlapping themes. We arrived at six final themes: *teacher's need students' needs to be met to experience joy*, *problem solving is a joyful process*, *joy can be found in the outcomes*, *relevant and meaningful curriculum designed for students' interests makes learning joyful*, and *students find joy by taking ownership*. The first

author verified all themes across all data sources.

Throughout the process, we resolved all disagreements through discussion (Hammer & Berland, 2014). For example the initial theme of *students finding joy by taking ownership* was separate from *relevant and meaningful curriculum designed for students' interests makes learning joyful*, and ultimately decided to keep independent because *Relevant and meaningful curriculum designed for students' interests makes learning joyful* was about what teachers needed to provide, and the *students finding joy by taking ownership* was about students' power to do what they wanted in the classroom. We also debated where to collapse the initial theme of *tangible learning* and concluded it should be part of *Relevant and meaningful curriculum designed for students' interests makes learning joyful*. We also debated whether *empower* should be its own theme, but decided to collapse it into *students finding joy by taking ownership* because students took initiative when they realized the power CS offered.

6.3 Joyful K-12 Computing

In this study we identified 6 ways K-12 teachers found joy in computing; *when students had what they needed, in community, in problem solving, in the outcomes of programming, in relevant and meaningful curriculum designed for students, and when students took ownership of their learning*.

6.3.1 Teachers need students' needs to be met to experience joy

Teachers dreamed of and shared stories where each individual students needs were met. Many teachers in the workshop highlighted how important accessibility was. In their ideal CS classrooms, they wanted to teach using programming languages that were accessible for all students. Teachers mentioned Wordplay ¹ and Quorum ², and asked for all curriculum to be "*natural language inclusive*".

And in designing the island accessibility was an assumption. The teachers designed the entire

¹<https://wordplay.dev/>

²<https://quorumlanguage.com/>

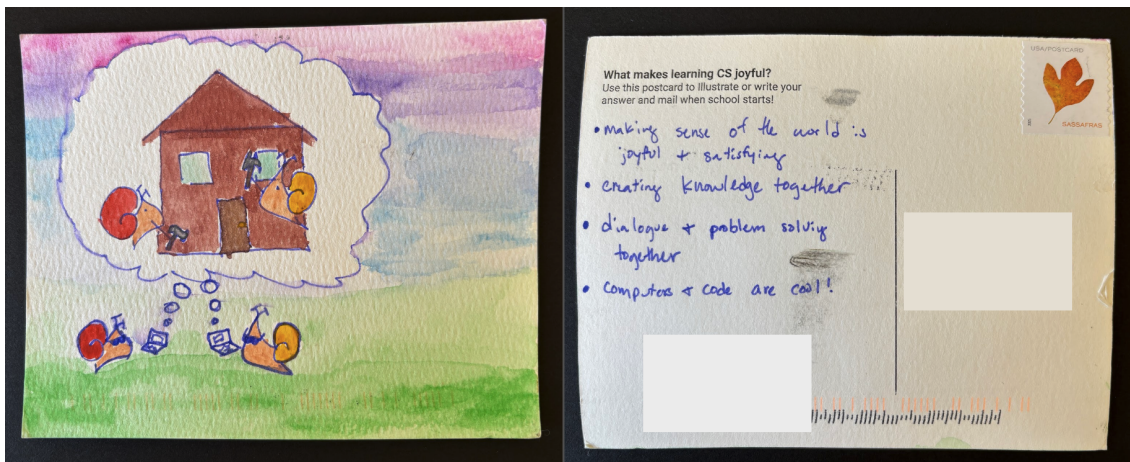


Figure 6.3: Postcard 3 is an example of the postcards teachers mailed in. They used the art tools to illustrate and share their thoughts and also often included a few sentences or short notes.

island to be *"Total access- whole island is like a giant accessible robot map"* where everyone was able to get everywhere and no one was excluded.

Also while designing ideal educational environments in the workshop, teachers expressed a desire for things that required resources like, *"smaller classes"* and curriculum. Teachers noted the types of curriculum they needed; curriculum that covered a *"wide range of topics + projects"*. They also asked for *"curriculum is inclusive and adaptive to the needs of all learners"*, was *"peer reviewed, rated by educators,"* and that had *"flexible starting points for students (block or text-based)."*

Additionally in the workshop, teachers wrote of a need for physical resources like *"wifi"* (Figure 6.1), or that *"each student has a device w/ fun, alternative flexible seating options"*. Air conditioning came up surprisingly often. AC was required, not just for student and teacher comfort, but as one teacher in the workshop discussion mentioned, *"A working air conditioner. not always there. not always there. It's been a big problem in my school recently."* and another teacher shared in response, *"the computers die because they overheat too."*

6.3.2 Joy is found in community and relationships

Community, collaboration, and relationships came up as an important component of joy in each of the data sources. A teacher wrote on postcard 3 how joyful it was to be *"creating knowledge together* and to, *"dialogue and problem solv[e] together"*, as seen in Figure 6.3. Another teacher noted the importance of collaboration and relationships between peers when they wrote.

" When my students started their own discussions about the work and quickly responded to their peers, it brought joy to me because my students were enjoying the lesson. " – postcard 5

In the workshop discussion, another teacher shared how they worked to make sure they were: *" incorporating like a healthy amount of like social interaction.* And then went on to describe how they felt like they were going against the flow.

" I feel like I what I've seen like a mistake that like CS teachers kind of make sometimes. like making like, you know, you focus on like the computer part of things, but you don't really focus on like that peer to peer interaction, which is like that's such like a variable, like joyful experience " – workshop discussion participant

They viewed peer interaction as an important part of learning computing and making it joyful.

Teachers also shared that their relationships with students were an important part of a joyful CS experience. A teacher wrote on a curriculum design poster that it was important to have relevant, reliable curriculum so that they could, *" focus on relationships, extra support."* In another example a teacher responded to the postcard prompt digitally, writing that joy in computing came from *"The ability to create things and connect with students through that creativity."*

In the workshop discussion, a teacher mentioned how it was important to connect to the broader community so that students were *"doing feels purposeful and it might like solve something bigger, it can connect to the world."* They mentioned that it did not need to be world changing work, it just needed to be purposeful and that *" sometimes it's just silly."* In the workshop posters, teachers wrote about another way to connect to community by creating learning environments

that were *"multi-generational"* and that students had *"relationships with mentors"* who were *"culturally relevant"* and *"role models"* (see Figure 6.1). It was important that students had the opportunity to connect to older students and people working in industry so that they could *"see it to be it."* Another teacher figured out a way to do that by inviting students back to their classroom:

" alumni that I've taught come back and talk to a group of students about their journey in CS. What are they doing now, and how did they get there? It's delightful, as I get to connect with old students and also inspire new ones." – November digital postcard

Teachers also mentioned that they need community too, in order to become better teachers. They found it important to connect and found value and joy in *"Meeting fellow CS teachers and learning new things. Now really excited for this project!"* at the summer conference. Another teacher shared in a beginning of the year postcard that they found it joyful when,

" All the CS teachers are coming together at the beginning of school is always fun! We had a teacher share about "six bricks" for teaching K + 1 about teaching computation thinking. We <3 it!! " – postcard 2

6.3.3 Problem solving can be a joyful process

Many teachers shared the process of working on CS was joyful because CS was like a puzzle. A teacher wrote on a digital postcard that they found CS joyful when students were able to *"Discover the Joy in Computing – where logic meets creativity, and every problem is a puzzle waiting to be solved!"* In another digital postcard a teacher shared:

" I am joyful when I see others solve problems they are encountering because I notice satisfaction and gained confidence through the time invested to find a solution. Through computer science, students, teachers, and others are able to solve problems."
– October digital postcard

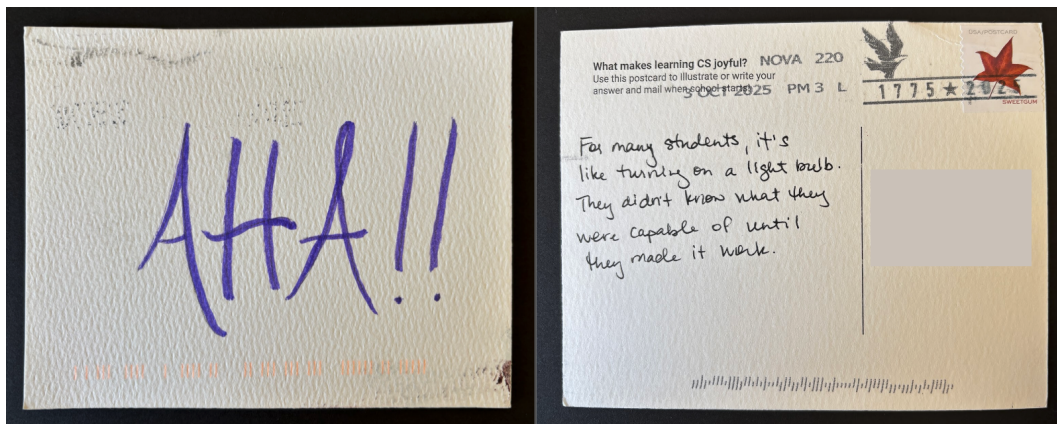


Figure 6.4: Postcard 4 shares a teacher's perception of the AHA moment that so many students find as a joyful outcome of studying CS.

Another teacher wrote in December in a digital postcard that they noticed the way CS classes created opportunities for students to unlock the tool of CS and experience joy while experiencing, *"the magic of transformation, like turning abstract problems into functional solutions, or finding your passion for problem-solving and helping people."*

Teachers noted conditions that were required for joyful problem solving. And workshop teachers wrote on the pedagogy poster that their ideal computing environments were ones where *"group problem-solving that doesn't feel like 'groupwork'"*, and *"use[d] real-world problems"*. In another example, a teacher emphasized that problem-solving was joyful when there wasn't pressure to solve the puzzle, writing on a digital postcard: *"When students get something to work! When they can solve a fun puzzle. When the pressure of time and grades is taken away"*

Teachers often emphasized that the hard work of problem solving was in service of an aha moment, or that moment when you finally figured something out. One teacher wrote about this on a digital postcard from December: *"There are so many different ways to solve a problem and it is very exciting and satisfying to solve a problem you've been working on!"* and another teacher wrote on a digital postcard that, it was joyful to watch when students were *"solving interesting problems, getting something to work, [and taking] opportunities to be silly."*

6.3.4 Joy can be found in the outcomes

Many teachers shared they experienced or witnessed joy in the outcomes of projects, for example, when students got code to work particularly after a period of struggle or frustration. They often referred to it as an *"Aha moment!"* as seen in Figure 6.4, or when students understood something as a *'lights-on' as joy* moment. Teachers also mentioned the importance of reflection and taking the time celebrate students success. Teachers wrote on the pedagogy poster from the workshop that it was important to create *"lots of opportunities to share successes"* and incorporate *"frequent celebrations"*. In the workshop discussion a teacher shared that it was important that

" once [students have] gone through it, like creating those moments of celebrate like, you, you know, that's the struggle itself in getting through that is the celebration."

– workshop discussion participant A

And another teacher responded with the importance of highlighting successes publicly sharing:

" So it's also almost like the kids feel proud of what they've done.... I feel like making that celebration social and like across the whole community so that it is public. They're being publicly recognized, they feel like they're a part of that classroom of that community, of the CS space and they can see themselves in it a bit more, too."

– workshop discussion participant B

And that noticing students success publicly entangled students' positive CS identities. Similarly, in a digital postcard a teacher noted:

" Watching students create a digital product and find 'joy' in their accomplishment. I've had students who 'don't like coding,' who are very proud of the work they have done." – digital postcard

Regardless of students' perceived relationship with CS, teachers noted they were able to find joy by seeing the output of their work.

In all of the experiences of joy, teachers only mentioned the outcome of grades as a thing they needed to decouple grades and stress from CS for students to experience joy. Two posters from the workshop, the curriculum and pedagogy posters, mentioned that it was important there were *"no grades, just milestones"* and that they wanted work that was interesting enough to rise about the ever present student question, *"is this graded"* and *"is it good enough"*. Another teacher shared in a digital postcard that it was also important that students did not have, *"too much homework so they can focus on ideas and not feel stressed and bogged down."*

6.3.5 *Relevant and meaningful curriculum designed for students' interests makes learning joyful*

In all of the sources of data, teachers noted that it was important for curriculum to be designed for students. They described this using descriptions like *"engaging," "relevant," "real world projects"* and *"meaningful."*

Teachers noted that when students were engaged, they lost themselves in the learning, and learning was joyful. In postcard 5, a teacher wrote: *"Learning is joyful when all your students are participating and learning."* And a teacher in the workshop shared the importance of

" finding engaging vehicles to deliver content through, like a storytelling or like through games, like stuff that's like really meets like students where they're at and like kind of just like allows them to like immerse themselves in content. " – workshop discussion participant

Other teachers highlighted the importance of designing classrooms and curriculum to reflect individual student's interests and knowledge. For example, in the pedagogy poster, one teacher wrote, *"student grouping by interest and competency level not by age."* And in the workshop discussion we saw how when students' interests were reflected in the projects they were doing they were engaged:

" Not just cultural significance, but social impact or even environmental impact. So, where I live in Florida, uh. We do a lot of projects that are related to things like

climate change and things like that, but because that's something that people directly relate to and students tend to be more engaged in those. " – workshop discussion participant

Physical computing was one way that teachers found joy and engagement. A teacher wrote *Joy is a great connection for students in learning and especially when we bring in physical computin!"* in a digital postcard. Other teachers mentioned tools they used, like the micro:bit ³. The physical nature of some of these tools offered a concrete way for students to understand and see their own learning.

Teachers also mentioned the importance of including identity and intersectionality when designing for students' computing joy. In the workshop a teacher mentioned, *"We had something similar by students engage something that's more real world, and applicable or [for students] if it has some sort of cultural significance"* and another participant responded sharing, *"So for our identity, we kind of hit intersectionality"* which matched with what was written on the resource poster *Identity-focused activities seeing yourself in your life*. Teachers thought that students experienced joy when they saw themselves in the curriculum.

Teachers also emphasized the importance of providing flexibility and freedom to students. For example, in the workshop discussion a teacher reflected on how important flexibility was to them

" I realized almost everything that I kept going back to was about flexibility of like the flexible seated, flexible starting point um a variety of ways to access information."
– workshop discussion participant

6.3.6 Students find joy by taking ownership

We saw in the previous section that teachers worked hard to provide curriculum designed for students that offered students choice, in that hopes that students would take initiative, make choices, and unleash the power of CS and experience joy.

³<https://microbit.org/>

In the workshop, teachers envisioned that students would take initiative by designing and building the transportation on the entire (speculative) educational island. In their design, students were using the power of CS to create a space that was joyful, inclusive, and accessible. One workshop attendee shared this dream in the discussion:

" Basically the entire island has to be fully accessible, and not constrained to the classrooms. So there will be a transportation system built by the students... We've got an indoor classroom, full outdoor classroom. Then you have the island, of course, so this takes robotics to different levels. and we have multiple coding languages that are accessible with braille displays. We were considering multilingual learners, so we're we really created the map, and they can start with the tiny robots and then they could build up and go outside and there's a safe place to try this out." – workshop discussion participant

Occasionally, teachers also shared that viewed themselves as learners and students of CS, taking initiative by finding new ways to use CS. For example, in postcard 10 a teacher shared:

" I love when I can take something I learn and find new ways to use it. Not sure why I love it more in CS than realizing I probably do it a lot in other areas. Just more joyful in CS." – postcard 10

One aspect of *students taking ownership* was students recognizing they had autonomy and then taking advantage of the choices teachers provided. They would demonstrate that autonomy by exploring in unexpected directions, or completing work what was required for class. A teacher wrote about this in a digital postcard. *"In CS, we used Microbits for the first time ever and I had a student go above and beyond and code Tetris which we all downloaded and played!"* In the second example this teacher gave, students were using their free time and taking ownership of their CS learning by choosing to return and work on their own projects. *"My classroom is a safe place for students to come when they have free time throughout the day and build, invent and explore"*

Creativity was another way *student took ownership*. In this example, the teacher was thrilled at the way students were combining their own interests, teaching each other, and demonstrating their knowledge:

" My students are making apps... The assignment was to teach someone else about a topic that you know more about than anyone in the class. It is so fun to see the topics they come up with! One student made a cool app about burgers in our local city. Tomorrow in class we'll do a 'show and tell' looking at all the finished projects. It is so fun to see everyone support each other! " – digital postcard

Students aren't just taking ownership, they're also building community and helping each other. In the second example this teacher gave, students were using their free time and taking ownership of their CS learning by choosing to return and work on their own projects.

Finally, teachers mentioned the word "empower" or "powerful" as part of joyful CS. They shared examples of students realizing the things they could use computing to do. In a digital postcard a teacher wrote:

" The biggest joy in Computer Science is when you finally feel like you have actual power over technology... when you finally figure out that one misplaced semicolon or fix the logic on that one loop, and everything runs exactly the way it's supposed to? That moment is pure adrenaline. It's the best feeling ever, and it totally makes the hours of frustration worth it... But the absolute best part is getting to work on projects you actually care about, like building a tool for your school club or a dumb little game just for you. That freedom to create something truly personal is the ultimate win. " – digital postcard

For this teacher, the most powerful part of computing was when students were able to use their computing skills to do things outside of the classroom.

6.4 Discussion

In this study we identified 6 ways K-12 teachers found joy in both teaching computing and learning computing themselves. While teaching, teacher observed joy in their students (1) *when students had what they needed*, (2) *were problem solving*, (3) *had relevant and meaningful curriculum*, and (4) *when students took ownership of their learning*. Teachers also found joy in learning computing themselves, and saw that joy paralleled in their students when they described joy in (5) *outcomes of programming*, and (6) *in community*.

These enactments of joy contrast with prior works' findings of students' perceptions of broader computing education experiences. Joyful computing meant that students were working in *community* versus working alone (Wang et al., 2016). Students experienced joy when creating things using *curriculum that was interesting and meaningful* versus doing work that was mathy and boring (chapter 4. Additionally, teachers' identified joyful computing when students were supporting each other, which is contrary to the defensive and fear-filled climates often found in computing classrooms (Barker et al., 2002).

Moreover, teachers shared that when they valued individual students, they observed joyful computing. For example, when teachers recognized that students do not all approach the classroom with the same experiences, needs or backgrounds they were able to *ensure students' needs were met* and *teach a curriculum that is relevant, meaningful, and designed for student interests*. This aligns with prior work calling for relevant and meaningful curriculum (K. Davis et al., 2021) so that instruction is inclusive.

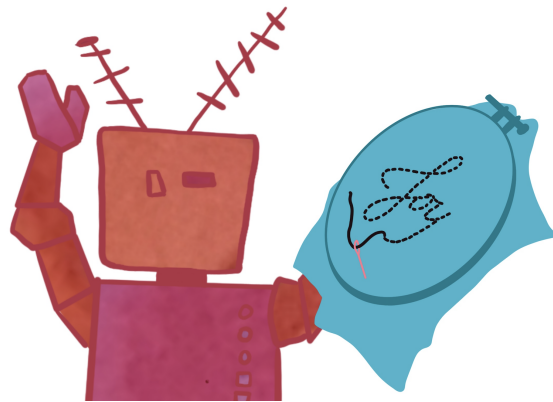
Feminist pedagogy asserts that relationships, community, autonomy, the valuing an individual's experiences demonstrate care and are important parts of education (hooks, 1994; Tamboukou, 2018; Woglom & Jones, 2024). The findings from this work align with feminist pedagogy. Teachers observed joy when students were *learning in community* by building relationships with peers and with their teachers: highlighting the importance of relationships. Students also found joy by *taking ownership of their learning*, which demonstrates the importance of autonomy. Additionally, teachers shared that when *students' needs were met*, for example, by having

a comfortable environment (e.g. airconditioning) and having *relevant and meaningful curriculum* that incorporated physical computing, both elements of an embodied experience, students found joy in computing. This aligns with the need to value an individual's embodied experience as part of feminist pedagogy (hooks, 1994; Tamboukou, 2018; Woglom & Jones, 2024).

There are several limitations with this work. We asked if our participants were teachers, but did not select for specific age groups or settings. The number of teachers who able to respond during the school year influenced our response rate, and the range of data. Additionally, our lived experiences as authors influence our interpretation of this work. If another group of teachers responded to these prompts, or other researchers conducted this analysis, they may identify different or additional elements of joyful computing. Additional work is needed to understand what joyful computing for instructors of different ages are, including elementary, middle, secondary, and university. Additional work is also needed to know how joy presents in different educational settings like informal contexts. Nonetheless, this work highlights the potential for practitioners to design computing courses and environments for joy, and the need for researchers to examine assumptions about existing pedagogy and how it influences students' experiences and learning. Researchers also need to examine if joyful pedagogies will help retain teachers in the classroom, which may help reduce the high attrition rates of teachers (Perez et al., 2025).

The nature of CS lends itself to joy: if we design for joy, rather than design against it. It is a field that opens doors for students to have the autonomy and choice to create things they care about; CS education gives students tools to build with. By combining feminist pedagogy with CS education, we can envision better, more inclusive learning for students and teachers alike. As feminist pedagogy encourages joy in the learning process (hooks, 1994) and CS offers the flexibility for students to incorporate their own interests and passions into their learning, the two together allow students to find joy and meaning in this work. If educators design computing using feminist pedagogies, students learn computing and learn joyfully.

Chapter 7

**JOYFUL CONTENT: AN EXPERIMENT WITH
COMPUTATIONAL EMBROIDERY**

This work was written with F. Megumi Kivuva as co-first author, and with the analysis support of Camilo Montes De Haro and Amy J. Ko.

A version of this work is published in the journal Computer Science Education (Everson, Kivuva, Montes De Haro, & Ko, 2026). In that article, I used the term Responsive Computational Embroidery (RCE) which was as close as I could get to describing my pedagogy and the pedagogy we used for this course and for another upward bound course I taught (Everson et al., 2022). When originally writing this article, I did not yet feel confident to use the phrase Feminist Computational Embroidery (FCE). As my understanding of my pedagogy has grown I realized that Feminist Pedagogy better describes the work we did. In this chapter, I have replaced RCE with FCE. I appreciate bell hooks sentiment: "we can live and act in feminist resistance without ever using the word feminism" (hooks, 1994, p. 62). I feel that reflects my experience learning more about feminism, feminist pedagogies and their intersections with CS education.

7.1 *Testing feminist pedagogies with a feminine medium*

I spent much of my time in my teaching position in a 6-12 school in Boston building, establishing, and running a makerspace.

In that space we had worked to incorporate many media used to support culturally responsive CS pedagogy. We had media like robots and games which have been found to enable students to collaborate and engage with peers and technology (Lewis Ellison, 2017; Li et al., 2023; Newton et al., 2020; Vella et al., 2020). We experimented with many programming languages. Researchers have also used programming languages to situate computational learning in the context of families of historically marginalized groups (Richard & Kafai, 2016; Roque et al., 2021).

Other researchers have engaged the approaches of philosophy and citizenship to remind students that they are the creators of the future (Ashcraft et al., 2017; Pinkard et al., 2017; Vakil & McKinney de Royston, 2022; Yadav et al., 2022). Our global history classes incorporated many of these principles into their coursework. Desportes (2022) and Payne (2021) tied the expressive form of dance and used domain-specific languages for students to build belonging and technical skills. Others use media that require students to create artifacts, such as art, virtual reality, or digital and physical fabrication. These artifacts allow students to tie their existing expertise into learning computing (Boda & Brown, 2020; Gaskins, 2021; Worsley & Bar-El, 2022).

Culturally responsive pedagogy (CRP) has been used to engage youth with digital fabrics and textiles. One example of this is *e-textiles* (short for electronic textiles), where electronic components such as batteries, lights, sensors, and micro-controllers are embedded in and onto fabric or textiles. E-textiles have received a particularly large amount of attention in CRP due to their affordances in engaging culture through community (Y. Kafai, Searle, et al., 2014), fashion (K. A. Searle & Kafai, 2015), and students' lived experiences (D. A. Fields et al., 2018; Guridi et al., 2021). In that school had to fight to teach an e-textile course because the title 'Wearable Technology' would not sound difficult enough on college applications. After several rounds of back-and-forth with the college counseling department, I finally got the course approved with some obscure name that may have conveyed technical expertise to colleges, but might have

frightened off students who were already engaged and eager to be part of this course. I had to hunt down all of the students that were interested and personally encourage and invite them to enroll.

Sadly, COVID hit the first week of that course, and I had to teach a wearable tech course remotely— which because it is physical computing is very difficult to do in a very digital medium like Zoom.

I also advocated the entire time I was in that makerspace to purchase an embroidery machine, but was prohibited by colleagues because they were "too expensive," and because "no one would use them." I however, was convinced that *computational embroidery* (CE) was a promising way to engage students in computation by using textiles. Embroidery is a technique that creates patterns on fabric using a needle and thread. CE uses a programmable embroidery machine to embroider patterns onto fabric through automation. In the context of education, this means using a machine to embroider student-programmed patterns and images onto artifacts. Figure 7.1 shows an example of using *TurtleStitch*, a CS coding platform that interfaces with an embroidery machine, to embroider onto fabric.

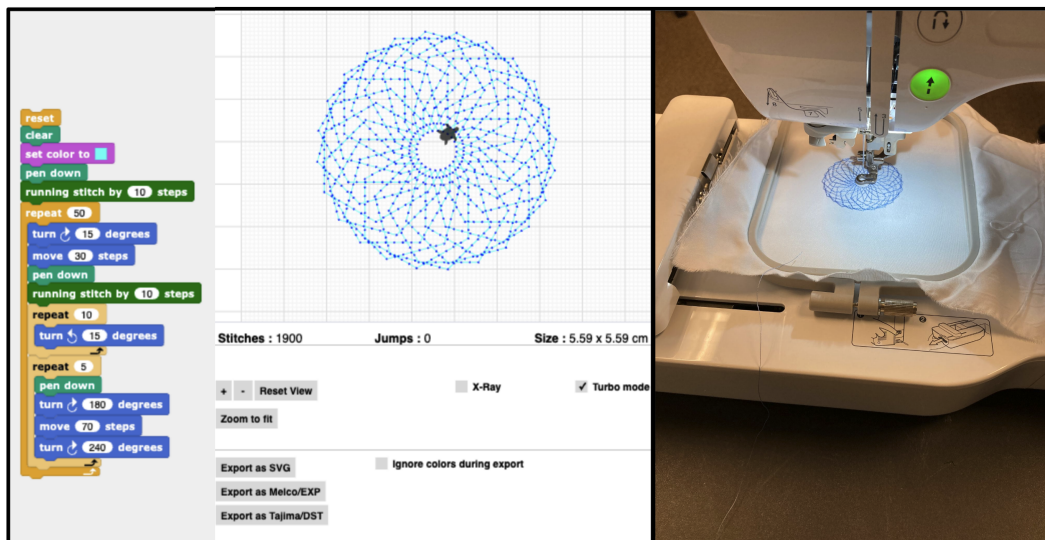


Figure 7.1: On the left is a screenshot of a student's TurtleStitch code from Week 3. On the right is the machine embroidering the work.

Recent work has explored practical ways of teaching and making with CE through workshops, demonstrations, curricular guides, and experience reports. (Y.-C. Lee & Albaugh, 2021; Schneider et al., 2020; Spieler & Krnjic, 2021; Spieler & Schifferle, 2022; Wolz et al., 2018, 2019a, 2019b). For example, in an experience report on this class that I taught with Megumi Kivuva and Camillo Montes De Haro, we found that weaving together CS, CE, hand embroidery, and culture, encouraged students to engage not only with CE, but also with culture and computation (Everson et al., 2024).

While this recent work demonstrated *that* CE is a viable medium to engage youth in culturally responsive pedagogy, it did not systematically examine *how* the medium shapes learning and teaching. It did not explore how this medium shapes students' computational learning or the effects on pedagogy and instruction. In this study, we define *shaping* as the educational and pedagogical implications of using the medium of computational embroidery in the classroom.

In a classroom, the instructor's pedagogy and the medium become interdependent. An educational medium is not itself a pedagogy, but will lend itself to certain pedagogies because of its properties and affordances. Of all the pedagogies available (i.e., direct-instruction, problem-based learning, project-based learning), an instructor makes pedagogical choices, which in turn affect the medium, the learning, and the instruction.

In this study, we selected culturally responsive and constructionist pedagogies paired with a critical perspective and care – as feminist pedagogies – to teach CE in the classroom. Together, we refer to these interdependent pedagogies selected for this course and the medium of CE as Feminist Computational Embroidery (FCE). The insights in this study are key in understanding both the pedagogical and technical opportunities that FCE imposes as well as its constraints and limitations in the classroom.

This study allowed me to test and better understand the interplay between feminist pedagogies, a traditionally feminine medium (CE), and computing education.

To this end, this paper answers two research questions:

- **RQ 1:** How does the medium of FCE shape student learning?
- **RQ 2:** How does the medium of FCE shape classroom instruction?

This paper contributes the following:

- Evidence for how FCE shaped student learning, illustrating (1) that FCE supports students to learn computational content while engaging their identities and cultures and (2) that students began identifying themselves as programmers.
- Evidence for how FCE shaped instruction, illustrating that CE and pedagogical choices were interdependent in supporting student learning.

7.2 *Related Work*

In addition to the theoretical frameworks of pedagogy covered in Chapter 2, this section also presents prior work on CE and E-textiles.

7.2.1 *Computational embroidery*

Previous work in computational embroidery (CE) has addressed how to make embroidery and what that experience is like in the classroom, but not how it forms learning and instruction. Recent work explored culturally motivated CE in an experience report, where they shared lesson plans and student work. They found that students were excited to engage in embroidery and teaching CS through the medium of CE empowered students to entangle their identity with their work (Everson et al., 2024). There have also been many maker activities and workshops at conferences to introduce embroidery. Some focused on computation and no programming (Schneider et al., 2020). Others introduced *TurtleStitch*, which is an embroidery-based programming language and platform (Spieler & Krnjic, 2021; Spieler & Schifferle, 2022; Wolz et al., 2019a, 2019b). In one study of a geometry game, researchers found that CE can help mediate technology, social crafting, and play (Y.-C. Lee & Albaugh, 2021). Although prior research suggests that CE in the classroom provides insights into how to utilize the medium and curriculum to engage students in cultural learning through this medium, we still lack an understanding of how the medium influences student learning.

7.2.2 *E-textiles*

In addition to a rich history of using textiles to teach technical content (Keune, 2022a, 2024), Many use E-textiles as a constructionist medium to broaden participation in computing. The *LilyPad* was instrumental in broadening participation by creating new communities for students to explore computer science (Buechley et al., 2008), and to build engagement and positive attitudes towards science with middle schoolers (Fey et al., 2022; Tofel-Grehl et al., 2017). Kafai et al. (2019) developed and evaluated a CS curriculum that teaches e-textiles with the goal of broadening participation in computing. This study found that e-textiles positively changed students' perceptions of computing, enhanced their self-confidence and creativity in solving CS problems, and improved their competencies in basic programming and circuitry, particularly among underrepresented groups. It also highlighted the need for deeper engagement with complex computational concepts. Although this prior work shows that e-textiles are an effective and engaging way to teach computing, it has not examined embroidery as a medium for learning.

Prior work in e-textiles demonstrates rich engagement with identity. Kafai et al. (2014) worked with Native American communities to introduce e-textiles in cultural contexts and found that they were able to build on community funds of knowledge and that students did work containing content they found interesting. When working with Native American boys, they found that students used computing for personal expression and that e-textiles became boundary objects, allowing students to make connections through computation. They also argued that educators need to value a variety of perspectives in the classroom (K. Searle & Kafai, 2015). When working with young Native American girls to make human-sensing sweatshirts, the team found that constraints gave students design agency, students had fun learning, and that these projects became boundary-crossing objects that traveled between the different spheres of life (K. A. Searle & Kafai, 2015). When embedding e-textiles in a classroom, researchers legitimized student expertise, thereby engaging all students (D. A. Fields et al., 2018). Another study found that incorporating e-textiles into formal classroom settings increased students' confidence in computing. They also found that teacher support led to a more equitable CS learning environment. (Y. B. Kafai et al., 2019).

In Chile, researchers collaborated with artisans to develop interactive collaborative patchwork pictures that combine quilting, embroidery, and e-textiles in a cultural heritage craft. They found this work supported communication and knowledge transfer (Guridi et al., 2021). Furthermore, a meta-analysis of all e-textile work called for drawing on students' funds of knowledge and experiences outside the classroom, noting that current assessments only capture one perspective of computational learning, rather than holistically assessing the intersectional nature of e-textile learning, especially when identity is involved. Thus, new assessment methods are necessary to holistically capture student learning (Jayathirtha & Kafai, 2019).

We also know that instructors and their pedagogy play a large role in the success of learning with e-textiles. Through an e-textiles professional development workshop, Nakajima et. al. (2019) found that rather than taking on the role of the traditional teacher, they found themselves being observers, facilitating student-driven learning. However, this research also found that teachers needed to be supported when teaching e-textiles, whether it be with affording materials or stress management. Support is needed to help teachers teach e-textiles in classrooms. Ryoo et al. (2015) explored the changes in pedagogy and teachers' transitions to inquiry-based methods, finding that teachers worked to guide students and practice the '*art of teaching*' and that these pedagogical shifts took time and community support. Additionally, instructors also need to understand the technology they are teaching to develop pedagogical content knowledge (PCK) (Jayathirtha et al., 2020; Shaw et al., 2020). These prior works show how student-driven pedagogy and teachers' confidence improve e-textiles learning.

7.3 Co-Constructing a Computational Embroidery Learning Space

To answer our research questions, we designed, taught, and examined an FCE class in the summer of 2023. This section details the course design.

7.3.1 Instructor positionality and reflexivity

Course instructors were the first three authors of this paper. We detail our positionality below:

Instructor Purple (the author of this dissertation) is a lifelong textile artist. She has 15 years of middle and secondary teaching experience, teaching math, science, and engineering. In that teaching experience, she has found that pedagogy that engenders constructionism CRP and feminist pedagogies is the most fruitful in creating rigorous, joyful learning environments for students. Her commitment to justice and equity in CS influences her work, and she works hard to listen to students to ensure their experiences are accurately heard and valued and that she holds an open and flexible mindset as she interprets data.

Instructor Blue is new to embroidery but finds the combination of computing, art, and culture in this project exciting. As a Ph.D. student who has taught computing in schools and other settings, they are committed to making computer science more equitable for everyone. Thus, they understand that their drive for equity and justice might influence how they conduct the class and interpret the data. By continually questioning their motivations and the impact of their background, Instructor Blue aimed to maintain an open and flexible approach, ensuring that their research and teaching were responsive to their students' needs and experiences.

Instructor Green has minimal experience with embroidery but has always been fascinated by its cultural relevance. He is a second-year undergraduate student and, therefore, closely understands the joys and aches of learning computing. He also has prior experience teaching an introductory computer science course. When teaching computing, Instructor Green always has equity and justice in mind. He is also driven to understand students' differences and to take an open and flexible approach to the class, students, and research.

It is also important to note our unique position as participant observers, as three of the authors were serving as both researchers and instructors. To mitigate this bias inherent to this dual role, we engaged in reflexivity (Patton, 2015). We were intentional about being constantly aware of our dual role, centering on constructivist epistemology which emphasizes the subjective relationship between researcher and participant (Mills et al., 2006). We recognized that our analysis was informed not only by our objective stance of systematic examination of data, but also by our subjective experiences teaching and of the relationships we built with students. Therefore, we surfaced tensions between these roles, and triangulated student and instructor data, to ensure

the authenticity of our thematic analysis (Patton, 2015).

7.3.2 *Context*

This course was part of a college preparatory program funded by the United States government. This program caters to low-income and first-generation students—those who would be the first in their families to attend college. Our local program, hosted at a large public university on the West Coast, serves around 125 students each year. The program is free for students to attend, and students receive a stipend for participating. Program staff recruit students through outreach programs within the local school district. Our course was taught as an elective course during summer school. Students ranked their top 3 electives and were placed in one of their choices when possible. Other electives include ballroom dancing, American Sign Language, and web design.

During the initial weeks of the course, students voluntarily provided their demographic data through an open-ended survey with all optional questions. Their responses are provided in Table 7.1. Twelve students enrolled in the course. At the start of this course, we explained the study's objectives, resulting in ten out of the twelve students assenting to participate and share their experiences and work, with parental consent also secured. One student among these ten participants had to depart from the class early, missing the final two weeks. The class had a diverse mix of students in terms of both gender and ethnicity (see Table 7.1). Students chose their own pseudonyms. The university's institutional review board reviewed and approved all research plans prior to research, and the research was deemed exempt.

7.3.3 *Tools and techniques*

In this class, students used the **TurtleStitch**¹ platform to design, and then exported the program they designed into one that the embroidery machine could read.

Embroidery Machine For this course, we used a hobbyist embroidery machine (Brother SE600). The machine was selected for its flexibility and affordability (around \$300 US dollars).

¹<https://www.turtlestitch.org/>

Students selected thread and threaded it through the machine, chose fabric and loaded it in the machine's hoop to provide appropriate tension, adjusted the machine settings to set the embroidery location and size, and then watched the machine embroider (see Figure 7.1).

Students also **hand embroidered**. This was done entirely manually, stitch by stitch. There is a wide variety of stitches to create various patterns and textures. During the process, the fabric is held in tension by an embroidery hoop. You can see an example of both hand embroidery and CE in Figure 7.7 where the white flowers are embroidered by machine and the raspberry leaves and bees are embroidered by hand.

7.3.4 Pedagogy

The medium of FCE and the pedagogy are interdependent. These pedagogical choices, therefore, also shape student learning. FCE, by its nature, requires students to create physical artifacts; it pairs well with constructionism (Papert, 1980). It also pairs well with feminist pedagogies because it values the embodied experience of learners (hooks, 1994). Learning was practical, as they applied embroidery and computing. There were many opportunities to share their work with others through gallery walks, instructor feedback, and a culminating family showcase. We made the pedagogical choice to *co-construct* the course with students. This meant that we worked collaboratively with students to choose and create activities and assessments that catered to their interests and their learning goals, as well as those of the course. We did this by discussing ideas as a class until we reached consensus on what the goals and requirements would be. Initially, this required scaffolding; however, as the course went on students grew more confident in articulating their own learning goals.

FCE pairs well with CRP and feminist pedagogies, because students are making independent artifacts which lend themselves to flexibility in direction (Ashcraft et al., 2017; Charleston & Charleston, 2014; Cooke et al., 2019; Y. Kafai, Searle, et al., 2014; Ladson-Billings, 1995; Nakajima & Goode, 2019; Scott et al., 2015). These pedagogies encourage students to explore and express their interests and identities. We worked to create an environment where students were free to express their interests and integrate their identities in the classroom. We provided

Pseudonym	Age	Grade	Gender Identity	Race/Ethnicity	First Gen	Other Identities
Eric	17	11th	He/Him	Asian Vietnamese	Yes	
Ricky Bobby	15	11th	Male	Hispanic/Latino	Yes	
Red	15	10th	Girl or Female	Hispanic and Latina	Yes	
Babo	16	11th	Male	Hispanic/Latino	Yes	
Bob		10th	Male	American Mexican	No	
Jasmine Milk Green Tea (JMGT)	16	11th		Asian Vietnamese	No	Another language at home
The First Ever Queen (FEQ)	15	10th	She/Her	Black	Yes	
Mimi	16	11th	Female	African/Somali	No	
Jellyfish ²	16	11th	Female	Chinese Immigrant	Yes	Immigrant
Chocolate Milk	17	12th	Male	Asian Chinese	Yes	

Table 7.1: Students self-selected pseudonyms and self-reported demographics of age, gender identity, race and ethnicity, first-generation college student status, and any other identities they wanted researchers to know. Students could opt out of any question.

students opportunities to learn about the cultural traditions of embroidery and use them in their projects. We hoped that by letting students focus on what they were curious about, they would feel more empowered, rather than worrying about grades. Lastly, we avoided asking questions with definite right or wrong answers. We wanted to show that there were many different ways to approach a problem, and no single method was necessarily best, as a pluralist epistemology supports (Turkle & Papert, 1990).

7.3.5 *Course structure*

The course was 6 weeks long, with 24 meetings and a total of roughly 20 contact hours. We organized each week around a topic where students and instructors co-constructed an assignment that included a hand or FCE deliverable and a reflection. Students collaboratively chose their reflection questions and what elements were included in the embroidery deliverable. Reflections often entailed writing about their process and what they were proud of or would have added if they had more time. Deliverables often included submission of a picture of the artifact they created and the code required. Table 7.2 provides a weekly breakdown of the course and the corresponding assignments with example photos for reference.

We began by first having students learn to hand embroider so they would have a foundational knowledge of the mechanics of embroidery before they started coding in TurtleStitch. Since one of the goals of the course was to explore cultural traditions of embroidery, in the second week, we took a field trip to the campus library, where students could check out books and search through the library's online catalog for resources about cultural embroidery. We taught students how to program in turtlestich for the remainder of the course. Students learned programming concepts like variables, loops, and functions. Students also learned how to export their code and upload it onto the embroidery machine to embroider their work. In the last two weeks, students worked on final project, where they elected to create work that included both computational and hand embroidery. The course culminated with an end of summer program banquet, where students proudly displayed their final projects for their peers, friends, and family. This event provided a platform for sharing their work with peers, instructors, and family members, celebrating the six

weeks of creative exploration and learning.

7.4 *Methods*

To answer RQ 1 and RQ 2, we collected student and instructor reflections about their learning teaching, and student work. We conducted a qualitative thematic analysis of the data we collected (Braun & Clarke, 2012).

7.4.1 *Data collection*

We collected weekly **student reflections** about their learning and creative process for their projects. Additionally, we documented **student work**, encompassing their projects and classwork (Table 7.2). Students submitted their work and reflections weekly for the instructors to review and grade for completion. We collected **instructor reflections** as daily individual reflections after each class period. Reflections detailed a summary of the class period, conversations with students, and any notable moments (e.g. challenges and triumphs). We selected these data sources intentionally. The student work allowed us to see what computational skills the students were learning and using in creating their artifacts with FCE. The student reflections provided insights into the students' experiences and affects learning with this media. The instructor reflections allowed us to triangulate the students' work and reflections.

7.4.2 *Data analysis*

We adopted Hammer and Berland's (2014) interpretive perspective on qualitative thematic analysis. They view qualitative analysis as a means of generating interpretive claims about data for subsequent investigation, rather than as a method for structuring data for quantification. Consequently, instead of presenting inter-rater reliability analyses and numerical data, we focus on detailing our analysis process and the interpretive disagreements that surfaced as we worked towards a collective understanding.

We inductively coded our data in three iterative rounds. During the first round, we all read the

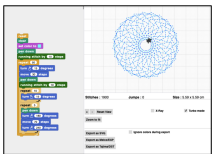
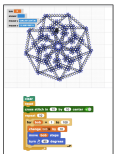
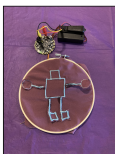
Week	Topic	Co-Constructed Objectives and Evaluation	Example Assignment
1	Hand Embroidery	Hand embroider three stitches and participate in a gallery walk	 Jellyfish
2	Cultural Traditions of Embroidery	Create a mood board with visual elements, external links, and reflective insights about traditions of embroidery; including cultural traditions if students wished	 Chocolate Milk
3	Introduction to TurtleStitch	Computationally embroider work of the students' choosing; possibly inspired by cultural traditions of embroidery; embroider their work with the embroidery machine	 Mimi
4	Students Choice: Continue Exploring TurtleStitch	Computationally embroider work of the student's choosing, embroider their work with the embroidery machine	 Babo
5 & 6	Final Project	Computationally embroider work of the students' choosing using TurtleStitch, the embroidery machine, and hand embroidery; integrate all concepts learned throughout the course	 FEQ

Table 7.2: Weekly course breakdown with each week's topic, co-constructed assignment, and an example of student work. Students began learning about hand embroidery and the cultural traditions of embroidery. Students then learned how to program with TurtleStitch and use the embroidery machine.

pseudonymized student and instructor data and surfaced themes about how FCE shaped student learning and instructor teaching. The next coding round consisted of identifying significant statements and linking images of embroidery projects and code that students and instructors reflected on. We also made affinity diagrams to surface disagreements and captured them in a memorization document. From this, we created a codebook of themes that emerged from the significant statements. In the third round of coding, we conducted a verification round where we re-coded all the significant statements with the themes in the codebook. To do this, the three researchers divided and re-coded all of the student data, each piece of data being coded by two researchers. Because the researchers were also the instructors of the course, for the instructor data, we did not code our own data. For example, Instructors Purple and Blue coded Instructor Green's reflections. At the end of round 3, we began to surface claims about the data through discussion. After coding, all researchers met to discuss codes and reach an agreement and made note of disagreements in a memoization document. Then we used these codes to develop claims about the data as seen in Table 7.3.

During the coding process, there were a small number of disagreements. Some of them were about the inclusion and exclusion criteria for specific codes. In these cases, we discussed the code's meaning in depth until we all agreed. We then updated the codebook to reflect any changes or certifications. Other disagreements were about the context of student quotes. We worked to balance interpreting the data within the context of the relationships we built with students while also not not inferring from student statements. We resolved each disagreement by surfacing the tension and deciding whether or not to include the data. We discussed as a group until we reached a consensus (i.e. all researchers agreed on a shared interpretation).

7.5 How FCE shapes learning and teaching

Our analysis revealed 5 themes and 9 claims regarding the ways that FCE shaped learning (RQ1) and instruction (RQ2). The themes and claims of our findings appears in Table 7.3.

Theme	Claim
FCE and learning	Students understood the strengths and weaknesses of FCE to learn computing and create the work they wanted(RQ1)
	Students used TurtleStitch to learn computation as a tool to do the work they wanted (RQ1 + 2)
Identity and Culture	Students used FCE to entangle their identities and cultures in their work(RQ1)
Meaningful Work	Students were learning computing to do work they were proud of and that they cared about (RQ1)
Growth	Students wanted to continue learning FCE and CS; and made plans to do so (RQ1)
	Students had identity shifts, and saw themselves grow as 'programmers' (RQ1)
	FCE required instructors to develop pedagogical content knowledge. (RQ2)
Instructional Implications	Hand embroidery established concrete foundations for future computational abstractions (RQ2)
	The environment designed for FCE allowed students to relax, learn to program and to lean into their programming identities (RQ2)

Table 7.3: Themes and claims.

7.5.1 FCE and learning

As a media, FCE shaped student learning by allowing students to create work they wanted to, and the programming language required for FCE allowed students to explore and learn computational skills.

Students understood the strengths and weaknesses of FCE to learn computing and create the work they wanted

Students embraced the challenge of using the FCE pipeline. This pipeline required them to write code in TurtleStitch, save and export the machine code, move to the embroidery machine, correctly thread the embroidery machine, and watch the stitching process to make sure no physical complications arose.



Figure 7.2: Red holding up the finished boba tea cup – looking like she wanted it to.

Students were able to use FCE to create exactly what they wanted. Like Red, who meticulously designed and programmed the cup of Boba seen in Figure 7.2. She reflected *"My favorite part was watching the work being finished and looking the way I wanted it to."*

Students were able to identify and use hand embroidery and FCE according to each media's

affordances. By the end, they were able to best combine the media as Mimi reflected:

"I think there is a lot of similarities between embroidering and the turtle stitch program but I think turtle stitch is a lot more easier and less time consuming as well as more precise." – Mimi

During the first week he was using TurtleStitch, Chocolate Milk was beginning to identify the limitations of TurtleStitch as compared to hand embroidery when he wrote:

"I don't know if I haven't figured out the program or what, but there's some stitching methods that's like not included in the program like, french knot or chain stitch." – Chocolate Milk

By the end of the course, Chocolate Milk really understood when to use which medium.

"Turtlestitch is best used for outlining your design and it isn't good for filling in. It takes more thought when you're coding too so I feel like if you're making something complex it you can save the effort of hand stitching." – Chocolate Milk

In the last week, Chocolate Milk discovered how to import images and translate them into embroidery, combining the best of FCE and hand embroidery. After he machine printed an outline, he embellished it with elements of hand embroidery to get the exact fluffy corgi butt texture he desired (see Figure 7.3). Chocolate Milk's corgi is just one example of how students impressed instructors with the ways they found to create work that they found personally meaningful.

Instructors noted that students shifted from initially using the machine only when prompted to expressing a desire to understand the entire process and combining their course learning with previous experiences and skills. After jellyfish had spent an entire class successfully debugging the embroidery machine by disentangling thread trapped in the machine, Instructor Purple shared:

"[I] spent a lot of time working with [jellyfish] who ran the machine and debugged it. We chatted a lot about how physical debugging works" – Instructor Purple

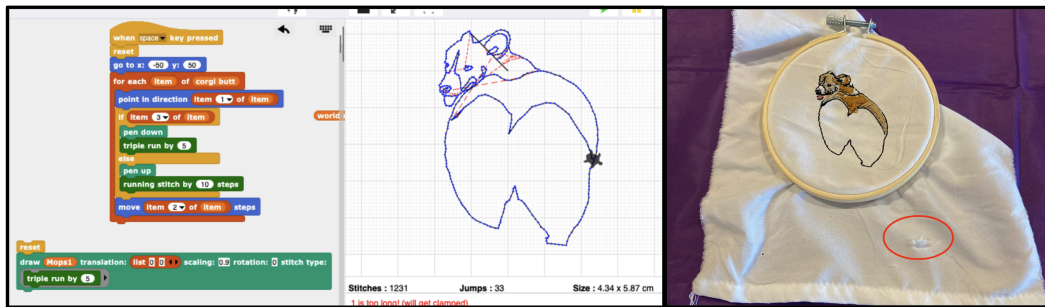


Figure 7.3: On the left is a snapshot of Chocolate Milk's TurtleStitch Corgi, On the right was the work he shared at the end of term program Banquet. The red circle highlights a stitch Chocolate Milk had discovered that day, and was intending to use to fill in the corgi's rear end after the end of the course demonstrating his fluency using both machine and hand embroidery techniques.

Jellyfish became the class expert and was the first person to jump up and check the machine if anything went wrong. Figure 7.6 shows jellyfish carefully watching the machine, ready to help, after debugging, fixing, and re-threading a previous snare. Students took responsibility for the machine and ensured it was capable of supporting the work they wanted to do.

Students used TurtleStitch to learn computation as a tool to do the work they wanted

TurtleStitch, as a programming language, organically created opportunities for instructors to introduce key programming concepts like variables, loops, and functions. On their own, students recognized the need to repeat chunks of code and requested that we, as instructors, explain how to make a function.

Throughout the course, students leveraged computational embroidery to learn how to program, and reflected on the things they had learned. A student who was very quiet in class, Bob, shared:

"I've learned alit [all of it] then just knowing programming because [now] I understand it better and learn about functions, variables and more." – Bob

Bob put together those functions and variables that he had worked to understand over the course with his final piece that he titled *The Silly Mad Goat* in Figure 7.4.



Figure 7.4: Bob showing off 'alit' put together with 'the silly mad goat' he made for his final project.

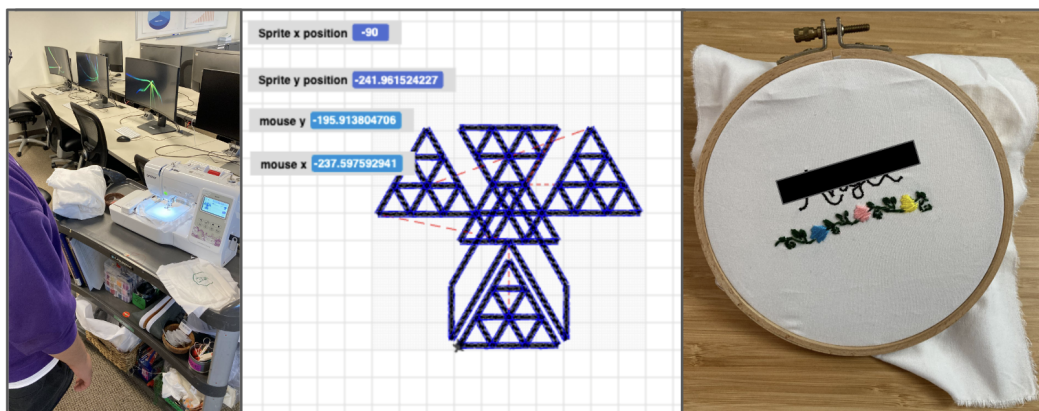


Figure 7.5: Ricky Bobby's work from left to right: Ricky Bobby running the embroidery machine, The middle image is the code for an Aztec pattern, Ricky Bobby's final work inspired by Mexican floral embroidery traditions.

Ricky Bobby enjoyed the challenge of figuring out how to get his code to do what he wanted:

"i liked the coding because coding is interesting to me but i also found it sort of difficult because it wasn't as easy to do everything that i wanted." – Ricky Bobby

Ricky Bobby experimented with traditional flowers when he noticed that Aztec patterns repeated, which made it a perfect place to experiment with learning to write code with loops and functions he wrote:

"I'm going to make a Aztec Pattern on turtle stitch. . . [I will need to use the code elements of] Definitely multiple loops because it has lots of repeating patterns" – Ricky Bobby

Once he figured out how to create these functions and loops, he created the pieces seen in Figure 7.5. In his final reflection, he shared that he enjoyed the freedom of expression FCE and TurtleStitch offered, and that his work represented him. He wrote:

"My favorite part of this course was the freedom we had as well as the second project i did because it represented me and looked really good" – Ricky Bobby

Instructor Blue noticed that students struggled with writing repetitive code to make circles and found it easy to introduce the concept of loops to help alleviate that struggle:

"We helped students debug in Turtle Stitch and troubleshoot their pieces. Students were frustrated that they were having to do things over and over again. We taught them about loops and I think they are getting a hang of it." – Instructor Blue

As students began to use functions, variables, and loops, Instructor Green reflected that they started integrating those more complex concepts on their own.

"Walking around, I can tell that students are becoming more proficient with the program. Their designs are looking more complex by utilizing more "advanced" blocks like functions, loops, and variables." – Instructor Green

TurtleStitch, as a programming language, allowed students to quickly learn coding concepts and then apply those concepts to new projects, creating artifacts that they cared about.

7.5.2 Identity and Culture

Students used FCE to entangle their identities and cultures in their work

Students were not just learning to code; they were using FCE to create physical objects that incorporated elements of their personal and cultural identities into projects. This was reflected in various aspects of their work, from mood boards inspired by cultural embroidery to coding projects that resonated with their personal stories. The students' engagement with coding was not just a technical exercise; it was a personal journey of self-expression and cultural exploration.

Students immediately incorporated their identities into their work and were eager to share personal details in the second week's gallery walk and mood boards. Instructor Green noted how students entangled their cultures and their identities when he wrote:

"Some designs were very culturally inspired, and even more mood boards had pictures of cultural embroidery for their inspiration in the upcoming projects. Angel embroidered a [local professional sports mascot] using a Mexican color palette (red, white, and green)." – Instructor Green

Figure 7.6 shows how jellyfish incorporated both jellyfish and circuits into the work they did. When sharing plans for this project, jellyfish also wrote about how they wanted to include culture and their clothes in a project:

"[My main concept is a] jelly fish. [I need these things this week:] chinese/japanese/korean culture: japanese jellyfish wind chime!!!! or fabrics i was thinking of [embroidering on] old clothes from childhood" – jellyfish

We found that when students entangled both identity and culture in their work, they did not differentiate between the two. While not a requirement of this course, many students chose to explore their own cultures.

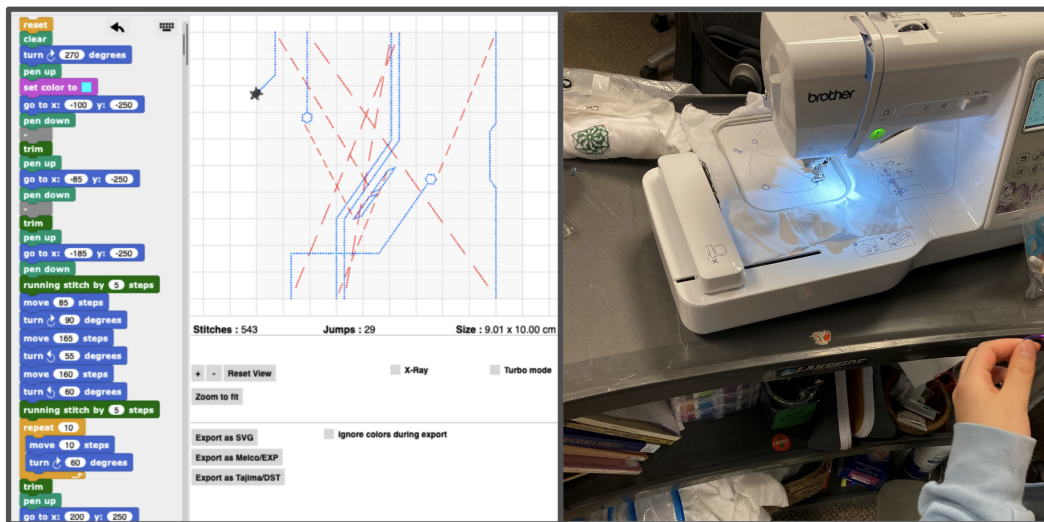


Figure 7.6: A snapshot of the code snippet and an image of jellyfish's circuit and jellyfish debugging the machine while embroidering their circuit.

Students mentioned specific cultures they wanted to explore or learn about. Often, these cultures reflected family traditions and heritage. Like Babo, who wanted to explore Hispanic embroidery and incorporate agave and cacti into his work.

"I learned about Hispanic embroidery and saw how many of the patterns are very vibrant and similar" – Babo

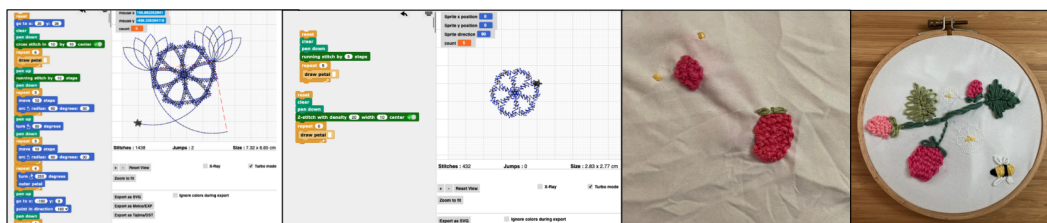


Figure 7.7: The evolution of JMGT's work- initially she started with a lotus flower, and then improved both her code, and the flower, and eventually included those flowers in her final piece along with raspberries and bees.

JMGT, who wanted to find Vietnamese embroidery, struggled to find resouFCEs at the library,

then expanded her search to include Hmong embroidery, and then broadened her search further. She shared the cultural elements she found that she wanted to start to incorporate into her work:

“During our visit to the library, I learned that many asian cultures tend to add specific meaningful symbols in embroidery pieces. So I am thinking of designing a lotus because it is supposed to symbolize purity, endurance and hopes for the future, this is because although lotus sinks in the night time, they always rise again in the morning and they are never stained with mud!” – JMGT

Figure 7.7 shows how JMGT started with a lotus flower in week 4. She then refined both her code and her vision for the flower in weeks 5 and 6 as part of her final project. Additionally, she iterated on scrap fabric to learn how to hand stitch the fruit and the bee shapes she wanted to combine with the machine embroidered lotus flower.

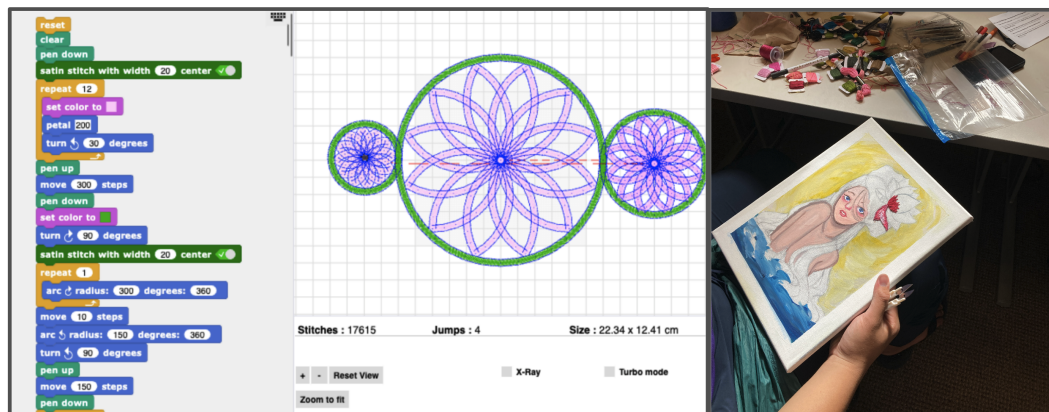


Figure 7.8: The left image is Eric’s Lotus code that he worked meticulously on. On the right is Eric working on his final project telling a childhood story by combining his love of painting and the embroidery techniques he learned.

Eric also used FCE to entangle identity and culture. Eric started by focusing on his culture and then worked to understand how to do that with FCE. He shared that he wanted to make: *“something that related with my culture a little bit such as lotus and trong dong detail. but more organized”* You can see Eric’s lotus on the left in Figure 7.8. Instructor Purple recounted a later

interaction with Eric, as he wanted to create several pieces as part of his final project to tell a Filipino story he had grown up hearing from his family. He compared it to Cinderella, to provide the instructors with context. Instructor Purple wrote:

“At the end of class [Eric] was telling us the Cinderella story from his culture and then ended telling us that he wasn’t very good at English.” – Instructor Purple

Eric enjoyed art and saw himself as an artist. He entangled that identity alongside embroidery by incorporating both mediums into his work, which you can see in Figure 7.8. He first sketched his plans in great detail, painted them on a canvas, and then embroidered over them, using FCE to combine his art and his story. Eric made several iterations of this final project to create exactly the image he envisioned. Each of these iterations had a lotus; some were hand embroidered, some were painted, and others were done by machine.

7.5.3 Meaningful Work

Students were learning computing to do work that they were proud of and that they cared about

Students expressed a sense of pride, not only in their final embroidered pieces but also in the process of creating them. This pride stemmed from various aspects of FCE: the experimentation and learning involved in FCE; the joy found in the journey of making the piece rather than just the outcome; and the personal satisfaction of seeing a project come to fruition. Instructors observed that students took immense pride in work that was not only complex but also held personal significance, often eagerly sharing their achievements and innovations.

In reflections, students shared being proud of their processes and approaches to their work. For example, jellyfish was proud that they did not take themselves too seriously, which enabled them to enjoy the process of programming:

“I am proud of just chilling and enjoying the process of learning to do these new stitches instead of worrying about perfection and only wanting the results instead of having fun during the journey.” – jellyfish

Often, beginning with a new programming language can feel intimidating, but Babo was proud of the risks he took in learning FCE, and the overall product he made when he shared:

"I am proud of me experimenting and actually doing it. It didn't come out too ugly but it came out decently I am proud of my stitches, the came out really nice." –

Babo

Student interest in coding grew throughout the course, and they persisted through challenges. When students triumphed over a particularly difficult chunk of code that enabled them to create a new stitch or pattern, they found instructors to show off their work. Like when Red sought out Instructor Blue:

"I told [Red] that I did not know how to do the stitch she wanted to do and instead of quitting, she figured it out and came to tell me!" – Instructor Blue

By the end of the course, students were no longer worried about meeting the requirements of the class; they were more concerned with learning the computational skills needed to do the work that was meaningful and that they were proud of.

7.5.4 Students' and instructors' growth

Through this work, both students and instructors grew in various ways. Students pushed themselves to work beyond the course requirements, become better programmers, and problem solvers. FCE required instructors to grow their knowledge of the programming language, the CE pipeline, and their CRP and constructionist pedagogies.

Students wanted to continue learning FCE and CS; and made plans to do so

All students were engaged and motivated to learn, program, and embroider beyond the requirements of any assignment. We found this last point particularly interesting because all assignments were co-constructed *with* students and the requirements of the course were set *by* students, not instructors. Students worked beyond the requirements for the course by engaging in work they

found personally meaningful. In a reflection about week 2, Instructor Purple mentioned that jellyfish had really enjoyed the library visit and had asked the librarian a number of questions. After class, jellyfish requested a list of books they were curious about that contained the history of Chinese embroidery, needlepoint, beadwork, and textiles that were in different libraries across campus. They wanted to incorporate these books and knowledge into their work.

Further, Instructor Green wrote about how different students were adding new mediums and materials to their final projects above and beyond what the assignment asked of them:

"Today the students continued working on their final projects. It was awesome seeing how each student personalized their project. One student wanted to incorporate simple circuits into their embroidery project, while another preferred to focus on hands-on art, using watercolor alongside hand embroidery. Students were comfortable exploring different mediums and how they can intersect with embroidery" – Instructor Green

It was also surprising to instructors to see many students request to take materials home and continue work after class.

In their reflections, students often shared examples that went beyond the requirements for the course assignments, and they sought to challenge themselves with new techniques. Red wanted to make a bird for their final project. She wanted a bird that had separate articulated and embroidered wings that would be able to flap and fly. However, she became so immersed with her cup of Boba (Figure 7.2) that she ran out of time to develop her bird. She shared, *If I had more time, I would like to work on another project like my original idea of making a type of bird.*

Students were motivated to persevere through challenges and eager to receive feedback. For example, in a final reflection, FEQ shared the frustration of only having an hour a day in class and only having four classes a week. She wanted to spend more time developing her final project, which was the moving robot seen in the last row of Table 7.2, using all the techniques she learned in class in novel ways:

"If I had more time to explore this class, I would have done a lot more projects and

tried, including both turtle stitch and hand embroidery at the same time and some of them.” – FEQ

Students also wanted to continue their work even after the course was over, through hand embroidery, more complex coding, or by learning additional programming languages. Students took action to continue their work. Babo took extra supplies with them at the end of class to continue embroidering:

“Hand embroidery was nice and cool, i really enjoyed that part because it was pretty calming to do. I want to keep doing it so i can keep doing it an then try to embroider other things.” – Babo

In a reflection, jellyfish shared that they wanted to figure out how to extend their knowledge beyond what they had explored in the course. They wanted to continue writing code and exploring other means of physical computation.

“some things I would like to learn next are perhaps other code languages and also how different code languages interact with different kinds of machines.” – jellyfish

Eric always took work home with him, and combined many media along with the embroidery and the FCE to make his art, as seen in Figure 7.8. During the final session, when asked if he was going to keep working on his painting and embroidery project, Eric enthusiastically responded “Yea! *why would I not?*” Students made plans to continue to learn and experiment with the tools and skills they acquired over the summer; some made plans to join robotics teams, others asked to have leftover supplies at the end of the course, and one student made plans to acquire her own sewing machine because she did not want to stop learning and experimenting with FCE.

Students had identity shifts and saw themselves grow as ‘programmers’

Even though students faced computational challenges, they were able to see their own growth. Students described themselves as programmers or coders. As instructors, we were careful not to

use language like *coders* or *programmers*, because previous literature noted that students could be reluctant to assume those identities (Lewis et al., 2016), so we found this choice of language in reflections particularly interesting.

In Week 4, students took charge of co-construction and expressed wanting to learn more computing concepts. Students elected to focus on their programming skills. Instructor Blue reflected on this successful co-construction:

“Students elected to learn more about turtle stitch, specifically variables, functions, and loops. Thus, we designed an assignment that requires students to create a project incorporating all three of these elements, as well as incorporating some of the cultural themes they explored in the library.” – Instructor Blue

Students surprised us with their eagerness to build programming skills; we were happy to focus on advanced topics. They also expressed interest in exploring various ways to interact with code to deepen their understanding of CS. Like Ricky Bobby, who wanted to test the limits of the system and experiment with increasingly complex code. *[I] want to see if i can learn to use the sensing blocks on turtle stitch.* Chocolate Milk recognized that he could utilize functions and recursion to make Japanese Sashiko patterns: *“I would try to mess with functions or recursion on turtle stitch to make sashiko patterns”* These sashiko patterns are often repetitive and detailed, lending beauty and structure to clothes and patching in Japanese traditions.

Many students wrote about how they took on this identity of programmer. For example, Red reflected:

“I have grown a lot as a computer programmer/coder because I did not know anything about it or what it meant by command but now I have been able to make the projects I have and learned about its variables, commands, what does what and more.” – Red

FEQ reflected on the relationship between her autonomy and how she’s grown as a programmer because of this course:

“And with turtle stitch, I learned the stitches were done a lot faster than being done with your hand, but with hand for me was more creative. My favorite part about the final project was being able to do anything I wanted. I have grown a lot as a computer programmer because I didn’t know as much about it before this class.” –

FEQ

Students were surprised by their programming growth, as their motivation was personal meaning, not just learning to code. They also noted mindset shifts that helped them view themselves in a different light. Like JMGT, who shared that she had the time and a project she cared about, to be patient with herself while programming. When she was able to successfully do that, her approach and mindset shifted:

“I think I’m less skeptical of coding now because before I had barely had faith that I could actually make something that turned out good. But I realized that it was something that just took time and patience.” – JMGT

jellyfish shared that they had previous programming experiences that did not allow them to build a strong computing identity. In this course and with FCE, they were able to iterate and make something that was strongly tied to their identity. They shared:

“TurtleStitch reminded me of my 6th grade coding class, which I struggled with a lot, and being able to revisit that again kind of rekindled my passion to learn code. Even though I had my difficulties with coding, being able to see my code create art by embroidery helped me find my intention in why I want to get better at coding...”

– jellyfish

Some saw their growth through the physical implementation of their code that FCE affords, and were surprised at their ability to successfully do things they perceived as difficult or complex. Like Chocolate Milk, who wrote:

"I learned to use block coding to embroider things. I had thought this sort of translating programming to real-life products would be much more complex to code." – Chocolate Milk

There are threads of growing student self-efficacy throughout their reflections presented in this paper. Self-efficacy was not a concept we introduced or discussed with students. However, instructors were familiar with this concept and closely monitored students' affects, attitudes, and self-talk. The instructors' reflections in this section triangulate the students' growth as programmers.

Students exhibited a wide range of prior experience and approaches to understanding programming concepts and shared a similarly diverse range of self-efficacy. At the beginning of the course, Instructor Green noted this:

"When the students were asked to pair up and annotate the example program, some groups struggled with how to start. The majority of the groups (usually each group had one person with prior experience) began splitting up the code into different sections to see what it did." – Instructor Green

Instructor Purple reflected on how the art of embroidery as an entry point seemed to allow students to worry about their work, rather than their previous experience or programming ability:

"Students felt comfortable asking questions and requesting more help on ideas they didn't understand, like variables. Yet, they were also very independent and are, to put it simply, quite good at turtletitch... My guess is that they are understanding the programming concepts well and are now focusing on creating what they want to create and rather than thinking so hard about programming. they are making art." – Instructor Purple

Initially, students did not see themselves as programmers. Student perceptions of their abilities did not always match with their actual abilities. For example, Eric deeply cared about his work,

was meticulous with his hand embroidery, and figured out ways to combine drawing and painting into everything he did. When we began using TurtleStitch, Eric pushed back on each coding assignment, often mentioning that he did not know how to do it, or that he did not know how to write code, when in reality, he was making beautiful complex patterns and programs (see Figure 7.8), while his reflections focused on the flaws in his work. At the end of the course, Instructor Green wrote:

"I also wanted to mention how good [Eric] is at using turtlestitch. On the first day, he was completely uninterested in the prospect of using turtlestitch– he thought he couldn't do it and should only stick to hand embroidery. However, today he used his creativity to fuel his desire to learn and explore turtlestitch. He even asked the person next to him for help and got inspiration from them." – Instructor Green

It was challenging to support Eric; he did not view his work as proficient and struggled to see himself as competent, despite instructors seeing solid work.

It seemed to instructors that students' perceptions of self-efficacy shifted over the course. By the end of the course, while co-constructing, Instructor Green noticed that students seemed to have increased their confidence with computing:

"When [Instructor Purple] asked the students to show, using their fingers, how comfortable they were with Turtlestich, most students raised 4 fingers, with fingers ranging from 3 to 5. I can also see this just walking around. The students are asking fewer questions ...and making more complex designs." – Instructor Green

Overall, instructors observed that most students viewed themselves as programmers who saw CS as a tool to create work they cared about, which aligned with student reflections.

FCE required instructors to develop pedagogical content knowledge

FCE required instructors to grow their pedagogical content knowledge (PCK) of the programming language, the embroidery pipeline, and their CRP and constructionist pedagogies. Teachers' PCK

is important in CS settings (Yadav & Berges, 2019). PCK is the intersection of teaching and content knowledge. In this FCE course, we needed to understand exactly how TurtleStitch worked and how students would use it. We needed to understand how the embroidery machine worked and how it failed, and we needed to know how to handle all of these technical situations through CRP and constructionist lenses, so that the learning was student-led and so that the classroom valued student choices, voices, and lived experiences.

In this course, we had three instructors with varied levels of computation and instructional qualifications and experience. Instructor Purple had served as a classroom teacher for more than a decade; Instructor Blue had taught computer science in a variety of informal settings, and Instructor Green had recently graduated from high school himself.

Instructors were relatively new to the language of TurtleStitch, were learning the quirks of the language alongside students, and developing their PCK in real time. In one example, Instructor Green shared:

“When helping [Bob] with trying to center some text in a circle, I noticed some innate issues with Turtlestich. It makes the process of figuring out and changing positions hard.” – Instructor Green

Over the course of the class, we built the PCK required to support students when programming, and we found fewer instructor reflections about those frustrations.

Instructors’ varied experience levels influenced their initial PCK, which was particularly evident when physically debugging the embroidery machine. In one experience where Instructor Purple was traveling, Instructor Blue wrote: *“I spent most of class debugging the machine, I felt like I was failing students.”* While Instructor Purple, who had more experience with this machine and physical pipelines, expressed the frustration at the friction required to share PCK:

“[Instructor Blue] couldn’t find the needles– so i jumped in and facetimed from Chicago. It felt hard to not be very helpful, and on a [phone] that was often face down. By the end of the course, each instructor had learned more about the physi-

cal pipeline required for FCE, and was able to better support both the machine and student work.” – Instructor Purple

Instructor Green grappled with his qualifications and legitimacy in the classroom and shared in one reflection at the beginning of the course:

“As for me, it felt a bit odd being an instructor in a room full of students, where a year ago, I could have been their peers. Even giving affirmations like “good job” felt out of place. It could have been because I wasn’t confident in my embroidery skills, and so I didn’t feel I had the right to say whether something was good. Maybe my attitude will change tomorrow when we start using TurtleStitch.” – Instructor Green

Instructors developed PCK for co-constructing assignments. At the beginning of the course, students were very worried about meeting the requirements, and instructors were worried about how to assure them that it did not matter. However, as students began creating projects that they cared about, we noticed that students were going above and beyond the requirements of assignments. Instructor Blue reflected:

“There’s a tension of expectations in co-construction. Who meets the bare minimum and who goes above and beyond. We are setting the minimum but how do students dictate when to go beyond that.” – Instructor Blue

As the course progressed, Instructor Blue’s reflections shifted, and they shared some of the nuances they were wrestling with of helping students to understand that they, as students, were setting their own expectations:

“ They are still needing some scaffolding; however students are more eager to share their thoughts [when co-constructing]. We were winding down our discussion of the assignment and one student asked if they were allowed to explore with colors and different shapes. How do we encourage students to be boundless in their thinking. Not just meet the requirements, but truly follow their heart and create what THEY

want to create? Kind of how doing what you love is the requirement.” – Instructor Blue

Throughout the course, instructors learned alongside students, developing their PCK to better support students.

7.5.5 Instructional implications of FCE

Pedagogical choices and the medium of CE are interdependent—the medium and its affordances shaped so the pedagogical choices we made as instructors, thus also the pedagogy shaped the learning and teaching.

Hand embroidery established concrete foundations for future computational abstractions

As instructors, we made the conscious decision to introduce hand embroidery before we introduced computational embroidery. This intentional choice allowed students to develop an understanding of how embroidery worked and the considerable effort required for each stitch before we transitioned to the automated labor of the embroidery machine. Had we not used FCE and used an alternate platform like Scratch³, there would have been no need to understand the structure and mechanisms of embroidery. We found that starting with hand embroidery also served as a way to build shared vocabulary and concrete representations that students would later use when exploring FCE. For example, to make a larger shape, like a cat, a student would have to make many tiny stitches to make the outline of the cat and then many more tiny stitches to embroider the cat fur. Often, these stitches are comprised of different colors of thread. Then the student can begin to embroider the eyes and the ear details. One could make the comparison between the stitches of embroidery and pixels in an image; many tiny pieces come together to comprise a larger picture. Because of this choice to start with hand embroidery, we found that later in the course, students were able to conceptualize big, complicated projects. For example, FEQ shared her understanding of how small components built bigger pictures by saying:

³<https://scratch.mit.edu/>

"I want to create something with small details that turns out into a cool big picture."

– FEQ

FEQ also shared that she enjoyed the fact that she was encouraged to be creative in this class and started to understand how the different types of embroidery afforded different applications:

"My experience with hand embroidery with a fun experience because it allowed me to be creative. And with turtle stitch, I learned the stitches were done a lot faster than being done with your hand, but with hand for me was more creative." – FEQ

Beginning the course with hand embroidery also provided students with a way to better understand the affordances of FCE, and to understand when to use it as discussed in section 7.5.1.

The environment designed for FCE allowed students to relax, learn to program and to lean into their programming identities

Because we started the course with hand embroidery, students spent the first week working with their hands and quietly chatting. We found that hand embroidery provided students with a meditative practice that fostered a calm and welcoming class environment. JMGT, who was very quiet at the beginning of the course, wrote:

"My favorite part of this course was hand embroidering but just be able to sit in a quiet space but not silence was cool. Not to mention, the silly check in questions we have sometimes and getting to hear peers thoughts." – JMGT

Throughout their reflections, both students and instructors emphasized the importance of a low-pressure environment in enabling them to engage with the material and take risks to explore new concepts. Part of this may have been influenced by pedagogical choices (co-construction, etc.), and part may have been shaped by the limitations of FCE.

Because there was only one embroidery machine, each student or team of students needed to work independently. They could not all do the same thing at the same time, so students were

less concerned with ‘keeping up with the class’ and more concerned with doing their work. In a rather profound reflection at the end of the course, jellyfish mentioned that the co-construction and class set expectations allowed them to take a step back and shift their goals from grades to learning, which shifted their self-efficacy:

“Since I didn’t have the pressure to do well on my first try, I got to actually learn and use critical thinking and planning to make variables. I also really enjoyed bringing my work home to work on it at home, hand embroidery is such a cozy activity and it allowed me to soak in the moment while I hand embroidered my project... This class really helped me to view coding and programming in a new sense. Because before, the time was limited and I just wanted a good grade, and it seemed like my code wasn’t up to the expectations. But now, I was able to learn to code in a more comfortable, less pressuring environment. This changed my perspective on my ability to code, and it’s giving me hope and it’s inspiring me to improve more on my programming.” – jellyfish

Due to the vulnerable nature of discussing culture and identity, instructors were particularly intentional about creating a welcoming environment. On one hand, instructors were very nervous about how quiet the class was. They were worried that students weren’t engaging with the material or that they weren’t interested. However, when the instructors started to read student’s weekly reflections, many students mentioned what a relief it was to embroider at the end of the day, and that the quiet class was actually calm and relaxing and how important it was to get to know each other, jellyfish even shared, *“We didn’t get to know each other until the end of week 4, I wish we had more time.”* The community built in the classroom was seen as vital to this work by both students and instructors.

7.6 Discussion

Our results highlighted FCE as a medium to engage students in computing. Students leveraged FCE to do work they cared about, work where they could see themselves in the pieces they

were creating, while also building new computing identities and making plans to continue their work. Our results also demonstrated the ways a medium can shape learning and instruction. Tools are non-neutral actors in education (Peppler & Thompson, 2024). FCE was a tool that allowed students to learn computing while engaging with their worlds, and experimenting with, understanding, and tinkering with both the concrete and the abstract representations at the same time (Blikstein, 2012).

7.6.1 RQ1 *How does the medium of FCE shape student learning?*

Students leveraged FCE to make exactly what they wanted to make, and they used the programming language, turtlestitch, to do the same.

FCE supported students in entangling their identities and cultures into their work. For example, jellyfish both chose the pseudonym *jellyfish* and also incorporated jellyfish into their final projects. Ricky Bobby incorporated Mexican embroidery and Aztec patterns into all of his pieces. This aligns with previous work using CRP (Everson et al., 2022; Scott et al., 2015). FCE as a medium lent itself to a culturally responsive context to effectively engage students. It gave physicality and validation to their culture and identity, driving motivation to continue learning. These findings are consistent with what we know about culturally responsive CS pedagogy: it helps foster positive student identity (K. Davis et al., 2021; Ladson-Billings, 1995). It also allows students to explore and regain what may have been lost or erased throughout history (Paris, 2012). As they explored their cultures and entangled their identities, these students also shifted from embroidery consumers, waiting for the instructors to set the pace and content, to embroidery creators, taking work home and working beyond the requirements of the course to make their work exactly as they wanted.

Students were proud of the work they created and pointed it out to us, learning programming to do meaningful work. Pride has also been observed in prior work in e-textiles (Jayathirtha & Kafai, 2019; Y. Kafai, Fields, & Searle, 2014), and meaningful work that is part of constructionism (Papert, 1980). Our work extends these findings by showing that FCE is also a viable medium for fostering meaningful work students are proud of.

Students made plans to continue learning CS and FCE beyond the course, and sought feedback that all align with substantial interest development (Renninger & Hidi, 2019). We saw student progress along the lines of Renninger and Hidi's four-phase model of interest development theory, which states that learners move from phase one of triggered situational interest to phase two of maintained situational interest, to stage three of emerging individual interest, to the final phase of well-developed individual interest (Renninger & Hidi, 2019). What we found surprising was that many students reached this fourth phase of interest development, where they were motivated to 'persevere through frustrations and challenges' and 'appreciate and actively seek feedback' in just twenty-four hours of class. Overall, students, for example, jellyfish and FEQ, expressed a desire to 're-engage with class content over time', demonstrating that they had reached the fourth stage of Renninger and Hidi's interest development (Renninger & Hidi, 2019).

Students had identity shifts and saw themselves as programmers. This is a novel finding as prior work has found that students develop *interest* (Lewis et al., 2016)), in CS, but seldom has it found that students take on CS as a part of their *identity*. This finding shows that FCE can allow students to further develop their self-identity as computer scientists.

7.6.2 RQ2 How does the medium of FCE shape classroom instruction?

Using embroidery and CE as the medium for this course allowed us to treat the traditionally feminized art of embroidery with the same technical reverence we held for computing. Starting with hand embroidery supported concrete to abstract cognition and allowed students to understand the limitations of the programming they were doing.

FCE shaped a positive educational environment. We worked to establish a calm, low-pressure, community-centered environment, and students felt comfortable engaging their identities and taking risks by experimenting with new technologies. The hand embroidery allowed us to engage with students about their lives while we learned new skills. Additionally, because each student worked independently, we wanted them to be able to voice their needs in assignments, so we co-constructed all deadlines and assignments. As a result, this meant students were able to engage in learning computing and FCE, rather than worrying about grades. These pedagogical choices

and the medium of CE were both essential in shaping the instruction.

FCE required instructors to develop PCK to use the programming well and to manage the physical pipeline. Learning both a programming language and how to manage a pipeline meant that things would go wrong in the classroom, which can be very vulnerable for an instructor. Often, instructors are positioned as experts in the room, and it can be very frustrating when things break or do not work. Because CS instructor self-efficacy has implications for student learning (Zhou et al., 2020), and that instructors' confidence, ability, and PCK are important in the classroom (Jayathirtha et al., 2020; Nakajima & Goode, 2019; Shaw et al., 2020), it is important to ensure that instructors have support to develop the confidence and PCK to work with FCE.

7.6.3 Importance of intentional pedagogy

Our work builds on hooks'(1994) concept of pedagogy as a non-neutral actor in education. We found our pedagogy was inextricably linked with the medium and decided to use the term FCE throughout this paper rather than just *Computational Embroidery*. In this work, we made intentional pedagogical decisions that could be made with almost any medium. Additionally, hooks notes that instruction is relational, and embodied. That means instruction and pedagogy will vary from teacher to teacher and from classroom to classroom. It is not enough to just examine the media students are learning with; the pedagogy of a learning environment must also be examined. The medium of computational embroidery was not a pedagogy, but it constrained what pedagogies were possible to use in the classroom, and what pedagogies would pair well.

7.6.4 Learning along with students

In this course, we built a small instructional community of three, learning from each other and with our students within our larger classroom learning community. Learning was not immediate for us as instructors and required experiences with student curiosity to develop skills to support student bugs and questions. These shifts challenged some of our previous notions of teaching,

as Instructor Green mused in section 7.5.4. This learning took time and community, much like previous research has found (hooks, 1994; Nakajima & Goode, 2019; J. Ryoo et al., 2015). We also needed support from each other to develop expertise and PCK to understand the embroidery machine, like teachers of other media (Jayathirtha et al., 2020; Shaw et al., 2020). Teachers need time and support to learn any new technology, and this inconvenient and perhaps expensive fact can be a forgotten part of technology implementation in educational environments.

7.6.5 *Physical pipeline*

In addition to introducing rewarding classroom experiences to students like motivation, identity entanglement, and boundary crossing objects (K. A. Searle & Kafai, 2015), physical pipelines introduce additional complications. There are additional compatibilities between computers and machines to keep in mind and understand, as well as physical debugging in addition to code debugging. Also, in this study, the transition from student intentions to digital representations to physical implementations did not match. What students thought was not the same as what the computer displayed, which was different than the tangible physical artifact, as Chocolate Milk mused in section 7.5.1. This study reinforces the idea that helping students (and instructors) navigate the complexity of physical pipelines. Also, students can become experts of the physical pipeline and real tools (D. A. Fields, 2009), and teach instructors, like jellyfish who became our physical debugger and fixed the embroidery machine.

7.6.6 *Limitations*

There are several limitations to these findings. Because of the nature of qualitative research, these results were specific to *this class* and *these students* with *these instructors*. The class was taught to 12 students, thus their work is only representative of their ideas. Also, the class was an opt-in elective, so the perspectives in this paper are from students who chose to learn CE, not from those who were compelled to. There were also 3 instructors teaching 12 students, when traditionally, in the US, one instructor teaches about 30 students with little to no help. Findings may have been

different had one teacher been responsible for helping many students and debugging the machine all alone. Moreover, although the researchers tried to analyze this data accurately and limit their biases, their lenses and experiences may have influenced their interpretations.

7.6.7 Implications

Given the findings and limitations of this work, there are a few ideas future research could address. First, further research could explore student interest development in computing and embroidery. Our findings showed quick interest and identity development in computing, with very few contact hours. Models like Renniger's (2019) model of interest development expect the level of engagement reached in our class to take months, rather than weeks; thus, prior work could examine what aspect(s) of the course led to such rapid interest and identity development. Additionally, were the findings here particular to embroidery and the constraints it imposes, or to the pedagogy and its constraints? Another avenue for future work might be to examine a class with a variety of creative tools and a subject focus (e.g. cultural traditions), or more embroidery machines, in order to examine the media against a variety of pedagogies. Additional areas for future research include (1) examining instructors' growth and change alongside student learning, (2) investigating whether it is better for students to simplify pipelines or to make pipelines more understandable and less of a 'black box,' and (3) exploring the interplay between pedagogy and media in CS classrooms.

Additionally, these findings have implications for how to create liberatory classrooms in CS that allow students to have positive experiences with CS, similar to prior work's findings that showed that students' perceptions of CS changed through learning CS with e-textiles (Jayathirtha & Kafai, 2019).

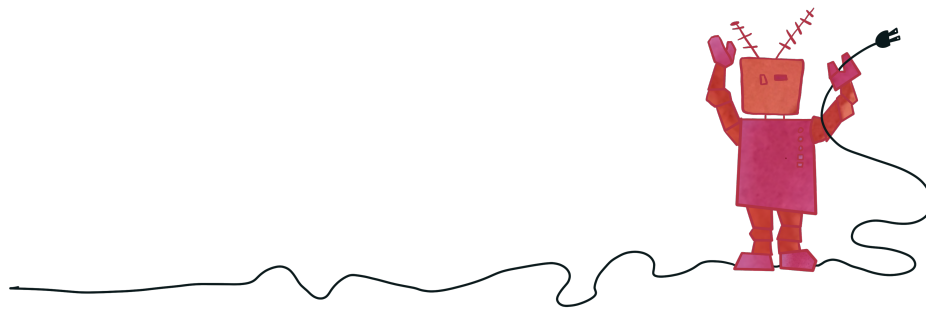
While this work points to important next steps in research, we believe our results have immediate implications for practice. Students were passionate and proud of their work and viewed themselves as computer scientists. We found it essential to include students' preferences. Students must be encouraged — but not forced — to bring their experiences, cultures, curiosities, and backgrounds to the work they do. This research shows that CE is a medium that provides

low floors, high ceilings, wide walls, as well as open windows, which allowed students to connect to their communities and cultures (Linn, 2009; Resnick, 2020).

This course did not look like a traditional CS course. We co-constructed the class around students' interests and identities, creating an environment where they could help curate their own learning experiences. We invite students, instructors, and curriculum designers to re-imagine what CS classes could look like if we imagined a future where students' voices are heard and integrated into course design. Computational embroidery was one way to ground these pedagogical approaches in cultural contexts; we hope future research and practice can explore FCE and other tangible media to better reflect students' identities, cultures, and communities.

Chapter 8

WHERE DO WE BUILD FROM HERE?



8.1 *What is joy in computing?*

In this dissertation, I shared five studies I conducted to better understand joyful pedagogies in computing education to support my thesis *feminist pedagogies encourage joyful secondary computing education*.

In chapter 3 I first explored assumptions of rigor in CS education, by interviewing undergrad computing students, secondary teachers, and post-secondary instructors. And found that many conceptions of rigor, particularly observations of rigor align with hegemonic defaults and are used to gatekeep students who do not match perceptions of what computing students should be. This reinforces the idea that current pedagogy in computing education creates miserable learning experiences for students.

Next, in chapter 4 using speculative design, I explored what pedagogies and environments students wanted in a joyful CS classroom. In this study I found that students desire pedagogy where the classroom is collaborative where everyone is accountable for their part, learning is

relevant and engaging and the experience of learning for students is pleasant, and where students have autonomy and choice in their educational experiences. These pedagogies align with the feminist pedagogy that values the experience of students and says it should be "pleasurable" (hooks, 1994; Tamboukou, 2018). When students were asked to dream of joyful computing environments, the pedagogies they highlighted were feminist pedagogies.

Then in chapters 5 and 6 I asked teachers and students where they found joy in computing classrooms using a combination of focus groups, design probes, and interviews. In chapter 5 students highlighted the feminist pedagogies like the importance of relationships in the classroom and care. They also pointed out when teachers prioritized learning and providing support while decoupling assessment from learning, and providing meaningful course work that was responsive to their interests as students. and trusting students to make choices in their learning through projects. When students shared experiences of joyful computing, they described feminist pedagogies in their classrooms and learning environments.

In chapter 6, teachers highlighted feminist pedagogies like prioritizing care, community, and relationships in the classroom. They also emphasized the importance of making sure the process of problem solving was joyful, and that the problems students were solving were part of relevant and meaningful curriculum. They also wanted to shift the ownership of learning to students. Teachers shared feminist pedagogies when they shared their experiences of teaching joyful computing.

Finally, in chapter 7 I tested the feminist pedagogies chapters 3 through 6 highlighted in a computational embroidery course. I found that the embodied experience of embroidering and the feminist pedagogies we used, and the environment created by those pedagogies supported students to engage their passions and identities in the work they did. Students used the medium of computational embroidery as a tool to create items they cared about. We also found that students wanted to continue their work beyond the course, and made plans to continue their technical learning. Students shared experiences joy in this computational embroidery classroom designed with feminist pedagogies. Harkening back to Roque's (2024) definition of joy, *"Not joy as positivity or happiness, but joy from transformation, that emerges from the challenges and*

struggles they encounter as they create projects,” it is very clear from the findings of this chapter that students were pushing through challenges and struggles they encountered to find joy in the artifacts they were creating. Students were discovering and learning concepts in order to make items that entangled their identities.

These studies together support my thesis that *feminist pedagogies encourage joyful secondary computing education*.

Joy in computing is when students are free to be who they are in the classroom. When students are able to bring their full identities into their work, and those identities are respected and valued. Where students are learning by doing work that is relevant and meaningful to them, and are able to focus on learning and growing in their knowledge of computing, and also their knowledge of who they are, and the world around them.

Joy in computing is the hope that the world could be better and the knowledge that we can make this world better and that we have the power to make this world better. Joy in computing exists in tandem and partnership with frustration at the injustice and inequity of the surrounding world. This coexistence is what allows students and educators to dream of better futures and because they see it can be better they enact and build a more just and kind world.

8.1.1 *Feminist pedagogy in CS education*

bell hooks writes that, *"Feminist politics aims to end domination to free us to be who we are"* (hooks, 2015, p. 118). Education is not a neutral act. It is a political one, and each form of education has its own implicit goals, the goal of feminist pedagogy (and politics) is to free individuals to be who they are by acknowledging the inequities and injustices of the world and through education actively work to construct a better world. This is counter to the masculine defaults of perfection, gatekeeping, and deciding if students are *good enough* to be computer scientists as surfaced in the examination of rigor in chapter 3.

In feminist pedagogy that means valuing the lived experience of the individual. There are two ways this work demonstrates that value; first, by valuing the knowledge and expertise students bring with them to the classroom, and second by actively creating joyful environments

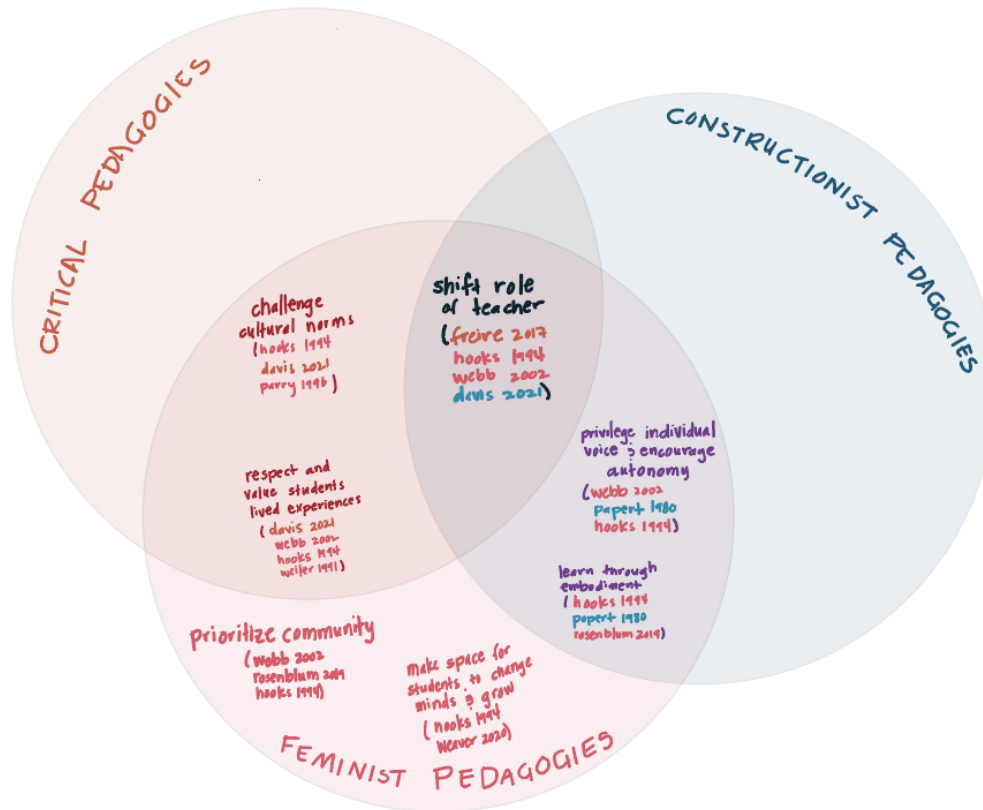


Figure 8.1: Joyful CS pedagogies are feminist pedagogies sit at the center of critical, constructionist, and feminist pedagogies. Feminism offers the relationality as well as the embodied construction of knowledge, while acknowledging the systemic inequities of the world.

and experiences for students. Culturally Responsive and Sustaining Pedagogies (K. Davis et al., 2021; Lachney et al., 2021) and also in indigenous pedagogies (Carruthers, 2024), and feminist pedagogy (Woglom & Jones, 2024) values that student knowledge and experience as an asset. Students and teachers dreamed of or shared experiences of curriculum that was relevant, meaningful and positively impacted the world around them in chapters 4, 5, and 6. Additionally, in the computation embroidery study, students created work that entangled their identities and was relevant to their interests.

The second of these two values presents in feminist pedagogies as embodiment. hooks writes about the importance of pleasure and enjoyment in the classroom (hooks, 1994). Feminist pedagogy values both the messiness of lived experiences and the experiences constructed in the classroom, and runs counter to the abstract absolutism and simplification that is so present in computing norms (Carrigan, 2024). In this work, I observed joy in embodiment in chapter 4 when students designed classrooms with comfortable seating so that they could sit with peers. I also observed that joy in Chapter 5 when students shared examples of joyful experiences where they had the ability to tie their passions into the work they were doing by designing interactive platforms or making games, and in Chapter 7 when students voiced how excited they were to entangle their identities in the embroidered pieces they were making.

To be who we are, we also need pedagogy that allows us choice and autonomy as part of freedom. This can be lived out in projects that students define in choose or content choice as in constructionism (Papert, 1980) or in feminism (hooks, 1994). This need for autonomy also applies to autonomy of body, in choices about how we engage in the classroom as teachers and as students (Flagg Young, 1900); for teachers this could be how they enact curriculum, for students it could as simple as how or where they sit. Students expressed how important it was to make choices about their work in order to have joyful experiences in each chapter where students were discussing computing work (chapters 4, 5, 7). In chapter 5 students also articulated examples of taking initiative to design their own joyful learning, or extend their learning beyond the requirements of their coursework.

Relationships, community, and trust are essential parts of building a joyful environment for CS education. It's clear from examining critical pedagogies in communities that relationships are strong and part of sustaining education (J. J. Ryoo, 2019a), however relationships are not mentioned as part of the core components in Culturally Responsive-Sustaining Computer Science Education (K. Davis et al., 2021) and Freire (2017) advocates for a shift in the role of the teacher from one as expert to one as co-learner, which also affects the relationship between students and teachers. Feminist pedagogy not only examines that student teacher relationship, it also emphasizes the importance of all relationships in the classroom (Blackmore, 2019; hooks,

1994; Weiler, 1991). In chapter 6 teachers mentioned how important and joyful it was that their students were collaborating with each other, and also with them. In Chapter 5 students shared how their teachers became fellow problem solvers and (sometimes gleefully) worked with them to discover solutions.

To build trust and strong relationships between teachers and students, it is important to distribute power in the classroom. Shifting the teacher relationship from one of authority to one of co-conspirator is one way to distribute power (Freire, 2017; hooks, 1994; Papert & Harel, 1991; Rosenblum & Harris, 2019; Webb et al., 2002) In chapter 4 students described accountability for teachers, students, and the class as a whole when they described their ideal joyful environments. Students described teachers who took on an approachable collaborative role in their joyful experiences in chapter 5. In chapter 7 instructors worked to de-center themselves in the classroom, and co-construct assignments, deadlines, and expectations in the course. Another way to distribute power in the classroom is to reconsider how educators assess students. Currently, assessing students sets up a relationship between students and teachers have different goals, one where students are trying to prove they are enough, and teachers are judging students and deciding if they are enough. In this work, students found alternative ways to demonstrate their learning, and were often not graded in traditional ways in chapter 5. Additionally, in chapter 7 students created the rubrics and requirements for their assessments.

Revisiting the pedagogy diagram from the the background Chapter 2 in Figure 8.1, it's clear that feminist pedagogy is not alone in offering liberation, however it is the pedagogy that includes the elements of joyful computing identified in this work.

8.2 What we need to build: future work

Feminist pedagogies in computing would require us to reexamine education from the ground up in order to create unbiased learning environments (Garr-Schultz et al., 2023; Margolis & Fisher, 2002). This work focuses on the classroom in particular as a unit of analysis. However, both smaller and broader structures influence education and learning. Feminist pedagogies in the classroom are not enough to change computing education. As a culture, we need to revisit policies,

school structures, administration, classroom sizes, individual interactions. Feminist pedagogies and praxes cannot be the icing we apply to the rotten cake, they must be the flour with which we bake the cake.

We also need practitioners to re-imagine computing education at every level. We need law-makers and policy makers to re-imagine how we think about schools at a cultural level. We need policy makers and educators to re-imagine the systematic level of how we sort students, how we grade students, and how we fund education. We need administrators and educators to re-imagine how we design the school day logistically, and pedagogically. At the classroom level we need educators and students to re-imagine the content and the pedagogy, and the at individual student level we need educators and students to re-imagine what schools look like for each student.

This means we need further research to study examples of feminist pedagogy at each of these levels. We need to understand how a change in policies and pedagogies affects individual students and their educational outcomes, not just grades, but their experiences, and the long-term implications of how these students move through the world and carry these pedagogies forward.

8.2.1 Measurement and Assessment in Tension with Feminist Praxes

In all of these studies, assessment of students causes a problem with the relationship between students and teachers. It shifts the relationship from one as co-conspirators and co-learners to one of adversaries. The student has to prove they are worthy to the teacher. Which is counter to the messages feminist pedagogies embody, that the individuals' voice, experience, and choice are enough (hooks, 1994).

Additionally, grading sets up competition in the classroom and creates a false scarcity making it feel unsafe for many students to take healthy academic risks. Theoretically there are an infinite number of abstract grades, but a numeric value applied to an embodied experience flattens the dimensions of that experience to a single dimension. This flattening of an experience will inevitably privilege some students' over others.

I've found some alternative ways to assess students' work by involving students in creating assignment requirements and the assessment process (Everson et al., 2022, 2024). However

further work is necessary to re-imagine a shift in the assessment system from where teachers compare and rank students to one where instructors partner with students to allow each student to learn at their own pace and be free to be who they are, while still learning.

The community also needs to re-envision assessment of learning. Current assessments privilege content over student's experiences because it becomes easy to compare students when you simplify learning to content. However, if students are finding joy, learning at their own pace, and pursuing their own interests, work is needed to understand what students are learning and retaining.

8.2.2 Redefining Rigor or Rejecting Rigor

As it currently stands, rigor has a muddled definition. No two people have the same definition, nonetheless rigor is used to define coursework and to gatekeep students. One way to redefine rigor could be to link it to the definition of joy in computing laid out in this dissertation; where students are free to be who they are while deeply engaging in the content they are learning through work that is relevant and meaningful to them.

However, if we redefine rigor, we risk losing the ability for the concept to evolve. We should always be learning and growing as a community and as people. If we define rigor in terms of joy as it stands now, we risk building a concept that will constrain us rather than free us. Where educators will

The idea that there is a single notion of successful learning is the problem. Joy will look different for each student. Rigor will look different for each student. Turkle and Papert 1990 argued that CS as a subject affords many avenues for exploration, yet CS as a field continues to value the same old exclusionary pedagogy, examples and practices. If we seek to hold any student up to a preconceived notion of what they should be, we are failing to see and value each student as they are.

The burden of rigor should fall on educators not on students. As language is evolving, I think that we should reject the term rigor, and in place precisely describe what students need to do to demonstrate mastery of the subject. However, if we fail to redefine rigor the field may continue to reinforce problematic definitions and gatekeep students with different backgrounds, experiences

and ideas. If we must, we should define it as: *students demonstrating their CS knowledge and skills by building things that make the world better*. Educators need to provide pedagogy and curriculum that allows students to explore and build.

8.2.3 *Feminist Praxes as a counter narrative to current implementations of AI*

I approach the topic of AI with hope. It offers a shift to the world we live in, like previous technological advances such as the printing press, the computer, and the internet. All of these technologies have changed how we share ideas, communicate, and work. All of these were initially surrounded with fear and loathing, yet we all use each of these previous advances everyday. AI could be a lovely chance to finally redo education instead of plodding along doing what we know does not work, and shift education to something joyful, relational, and beautiful. However, as it stands AI is a tool of capitalism and not humanity.

As it is implemented, AI is harmful. AI is built out of stolen property with the promise of replacing jobs rather than augmenting and making jobs easier (Al Naqbi et al., 2024). It is invested in and decisions about it are held by rich tech oligarchs, rather than democratically by people.

In education, students are using AI to replace work rather than do the work— often this is because of assignments are not meaningful, or because competition negatively affects a student's ability to engage. Instead of iteratively completing an assignment until they understand, students are penalized for the process of learning with low grades. Instructors create a false scarcity around grades where there is no scarcity of learning. My experiences teaching have shown me that every student wants to do well and learn. If they are not engaging with the course or the material, it is because something else is at play. Often its because they have too many obligations to do to invest the time they want into learning. Other times it is because they are afraid to try because doing badly would affect a fragile identity they hold. Other times it is because and authority figure like a parent or another teacher has communicated to them they are not good at something, or even worse that they are not worth investing in.

Feminist pedagogies in computing value first and foremost the human. It values each human

in the classroom and the lived experiences they bring with them. It values their position in society and the viewpoints on the world. Feminist pedagogies value the embodied experience of learning; learning should be fun.

Feminist pedagogies offer us a unique lens with which to approach computing education with AI much in contrast to what the corporate AI voices offer. These voices are offering to replace humans with AI, rather than recognizing the value of having humans in the process, and valuing the experience of humans.

If educators approach computing education with AI from a feminist lens we'll consider the implications of not only how we teach, but what we make, the process of how it is made, and how it affects the world around us.

8.3 Recommendation for joyful pedagogies and practices

Here I present some recommendations for joyful pedagogies and practices. See Table 8.1. They are all rooted in feminist theory, but I don't believe it's necessary to say we're doing feminism. We can just do feminism. I observed these practices in the research done in this dissertation.

8.3.1 Value individuals' contextual lived experiences by incorporating relevant and meaningful curriculum

As educators, we must value each learner as they come to us, what they bring with them, and for whatever they may become (Jasinski & Lewis, 2016). Educators must design for needs and interests in the classroom (hooks, 1994; Parry, 1996; Rosenblum & Harris, 2019; Webb et al., 2002), matching the findings from chapters 4, 5, 6, 7. This means that each incarnation of a computing classroom should vary depending on what students are walking through the door. As teachers, we should co-construct with them to make sure that they have relevant and engaging course content. Chapters 4, 5, 6, and 7 all share examples of the importance of relevant and engaging curriculum.

Feminist Principles in Education	Joyful Pedagogies	Description	Examples in this work
Recognize we come from a world saturated in sexism, racism, and oppression, and we will always be unlearning.	Design for curiosity and growth and challenge computing norms	Pedagogical practices must offer true freedom and action must match values; actions mean more than words. Encourage making mistakes, questioning assumptions, and considering new ideas.	When dreaming of joyful cs classrooms in chapter 4, students described an environment that directly countered their perceptions of what a CS class was. They wanted classes that were relevant instead of boring and mathy. Students described teachers who worked to make sure their learning was centered, and that mistakes were safe to make in chapter 5. In chapter 6 teachers dreamed of inclusive experiences where all students' needs were met, and everyone was able to learn computing. In the computational embroidery classroom described in chapter 7 we taught the course using a nontraditional medium, and centered what students were making instead of the concepts they were learning.
Value individuals' contextual lived experiences	Incorporate relevant and meaningful curriculum	Design each class around the students in the room, incorporate and value their interests and experiences	Students and teachers dreamed of or shared experiences of curriculum that was relevant, meaningful and positively impacted the world around them in chapters 4, 5, and 6. In the computation embroidery study, students created work that entangled their identities and was relevant to their interests.
Experiences are situated and embodied and should be pleasurable	Design for a pleasant embodied experience	One default of computing education is that it is miserable and requires a grind, instead design for joy and flexibility.	In chapter 5 students described joyful experiences where teachers made their coursework and their projects embodied and tangible, and worked to make sure that students were laughing and enjoying the course. Teachers described physical computing as a way they experienced joy in their classrooms, and described comfortable environments where all students' physical needs were met in chapter 6. Students built concrete representations of abstract concepts by hand embroidering, and worked to have a cozy, friendly environment at the end of the day in the computational embroidery class in 7
Privilege individual student's voices and opinions	Offer choice and autonomy	Offer students choice in content, co-construct assignments and projects, scaffold academic independence with students.	Independent studies and projects students described in chapter 5. Work students created in their final projects in chapter 7 were designed by students, and students incorporated additional mediums and pursued computational concepts to make work they wanted to.
Prioritize relationships and community	Design for collaboration	Work to incorporate collaboration between peers and with the educator, celebrate success often, and consider examining how students are assessed	Collaborative environments students described as their dream environment in chapter 4. Collaboration with peers and instructors students described as part of their joyful environments in chapter 5. Peer collaboration and care that teachers described in their joyful experiences in chapter 6. The "chill" and "relaxing" environment the class established in the computational embroidery course in chapter 7.
Distribute power in the classroom	Shift role of teacher	Shift the role of teacher from being "in-charge" to being "co-investigator" learning alongside students	In chapter 4 students described accountability for teachers, students, and the class as a whole when they described their ideal joyful environments. Students described teachers who took on an approachable collaborative role in their joyful experiences in chapter 5. In chapter 7 instructors worked to de-center themselves in the classroom, and co-construct assignments, deadlines, and expectations in the course.
	Decouple learning from assessment	Co-construct assessments with students, encourage mistakes, design for flexibility in deadlines and requirements	Students found alternative ways to demonstrate their learning, and were often not graded in traditional ways in chapter 5. In chapter 7 students created the rubrics and requirements for their assessments.

Table 8.1: Recommendations for joyful pedagogies in computing education and examples from this study

8.3.2 Design for a pleasant embodied experience

In valuing individuals, it is important to design for their embodied experience. Students' physical and emotional experiences matter. They should be joyful (hooks, 1994; Rosenblum & Harris, 2019).

In this work, I observed concrete and tangible examples and assignments as part of students' embodied experience. Sometimes, teachers explained abstract topics with examples built around the classroom as I discussed in chapter 5, or taught abstract concepts using physical computing 6. Other times, students turned their own learning concrete, and made projects into tangible physical artifacts, like with computational embroidery in chapter 7.

The process of learning, although sometimes challenging and difficult, can still be joyful (Roque, 2024). CS can be framed as a tool to do the things that students really want to do as in chapter 5. And the process can be joyful. In fact, it can support students' deep engagement and pursuit of complex topics.

8.3.3 Privilege individual voice by offering autonomy and choice

It's important that as instructors, we approach CS education with flexibility, and design for students' choice and autonomy (hooks, 1994; Webb et al., 2002). Rather than predetermining all content, and requirements, flexibility allows students to take ownership of their learning, like the examples of independent studies and projects shared in chapter 5, and the artifacts students constructed in 7. There is no way to anticipate the direction of students' curiosities, and it is important to respond with warmth, encouragement, flexibility and support. Educators must allow students to make choices and define their own learning. That is a goal of education- to scaffold students to become independent learners who are always seeing to become better people. Students need opportunities to become autonomous learners.

8.3.4 Prioritize relationships and community by designing for collaboration

It is important to prioritize relationships and community in CS education. This was found in each study in this dissertation. In chapter 5 it was exceptionally clear that relationships underpinned much of those joyful experiences that students described. Relationships are valued in feminist pedagogies (hooks, 1994; Parry, 1996; Rosenblum & Harris, 2019; Webb et al., 2002) and underpin much of learning by creating safe environments where students trust teachers and are willing to take academic risks, and teachers trust students to take ownership of their learning, and to pursue topics they want to.

8.3.5 Distribute power in the classroom

There are several ways to practically examine power in the computing classroom.

Shift the role of the teacher

The role of teacher in the classroom shifts to one of co-conspirator and co-learner (hooks, 1994; Rosenblum & Harris, 2019; Webb et al., 2002). Teachers and instructors act as coaches to support students in their learning as students described in chapter 5.

Decouple learning from assessments

It's important to decouple learning from assessments. The findings from chapters 4 and 5 emphasized the importance of collaboration over competition students shared how important it was to be able to communicate with each other and how much they enjoyed collaborating with each other and their instructors. Competition and judgment in classrooms make it hard to take risks and to engage in the vulnerability required to engage in learning (hooks, 1994). Co-construction of content and assessments, as described in chapter 7, is one such way to shift that relationship.

This shift in assessment does not mean that there should not be critical feedback. Open honest communication and critical feedback is an act of love (hooks, 2018) and essential for improvement and growth. However, grades rank students and create competition.

8.3.6 Design for curiosity and growth and challenge computing norms

This process of joyful teaching will never be finished. In the classroom, we will always be unlearning and discovering how we have been influenced by the computing norms and the broader culture of the world around us (hooks, 2015). As educators and humans, we will continue to discover new ways to love better, and to better design environments, structures and pedagogies – otherwise we become rigid, and create new forms of rigor that will replace the old forms and education will continue to exclude students.

Additionally, in computing education, the field is an ever evolving one. New programming languages and technologies will always emerge. Educators will have to adapt to best guide and support their students to become free to be who they are, and to engage in computing. In chapters 6 and 5 educators made references to the need to learn alongside students, and to learn new skills to support students.

8.4 There is no conclusion

Like the recommended feminist pedagogy of leaving room to unlearn and grow, as educators and humans, we will always have to work towards building a just future. Feminist pedagogy encourages joyful secondary computing education, building towards that just future. When instructors teach with feminist pedagogy they value and support all students in the classroom by respecting their prior knowledge and experiences, working to incorporate that expertise into the classroom, and by offering the students the freedom to pursue what they are curious about and by offering the students the freedom to become who they are. Teaching with feminist pedagogy also values the autonomy of individuals, and instructors will trust individual's choices. This trust between instructors and students is built through relationships and community.

These pedagogies run contrary to many current pedagogical assumptions and norms in computing classrooms. Currently, students perceived CS education as miserable, lonely, and hard (Carrigan, 2024; Everson et al., 2025; Wang et al., 2016), rather than joyful. Some students choose to opt out of computing rather than continuing in a field that also perpetuates norms of

exclusivity and abstraction that students shared in chapter 5.

To build a just future, our world needs all people building it, and joyful feminist pedagogies offer the possibility for a kinder more human future.

BIBLIOGRAPHY

- Agbo, F. J., Oyelere, S. S., Suhonen, J., & Adewumi, S. (2019). A Systematic Review of Computational Thinking Approach for Programming Education in Higher Education Institutions. *Proceedings of the 19th Koli Calling International Conference on Computing Education Research*, 1–10. <https://doi.org/10.1145/3364510.3364521>
- Ahmed, S. (2000). Strange encounters : Embodied others in post-coloniality. In *Transformations, thinking through feminism*. Routledge.
- Ahmed, S. (2023). *The feminist killjoy handbook: The radical potential of getting in the way* (First edition.). Seal Press, Hachette Book Group.
- Al Naqbi, H., Bahroun, Z., & Ahmed, V. (2024). Enhancing work productivity through generative artificial intelligence: A comprehensive literature review. *Sustainability*, 16(3). <https://doi.org/10.3390/su16031166>
- Ambrose, S. A., & Hornig, L. S. (1997). *Journeys of women in science and engineering: No universal constants*. Temple University Press.
- Arum, R., & Roksa, J. (2011). *Academically Adrift: Limited Learning on College Campuses*. University of Chicago Press. Retrieved August 25, 2023, from <http://ebookcentral.proquest.com/lib/washington/detail.action?docID=648124>
- Ashcraft, C., Eger, E. K., & Scott, K. A. (2017). Becoming Technosocial Change Agents: Intersectionality and Culturally Responsive Pedagogies as Vital Resources for Increasing Girls' Participation in Computing [eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/aeq.12197>]. *Anthropology & Education Quarterly*, 48(3), 233–251. <https://doi.org/10.1111/aeq.12197>
- Ashlock, J. M., & Tufekci, Z. (2026). Gender differences in computing interest: The role of social constructs in early paths. *Computer Science Education*, 36(1), 72–104. <https://doi.org/10.1080/08993408.2024.2433354>
- Association, C. S. T. (2017). CSTA K-12 Computer Science Standards, Revised 2017. Retrieved August 25, 2023, from <https://csteachers.org/k12standards/>
- Azevedo, F. S. (2013). Knowing the Stability of Model Rockets: A Study of Learning in Interest-Based Practices [eprint: <https://doi.org/10.1080/07370008.2013.799168>]. *Cognition and Instruction*, 31(3), 345–374. <https://doi.org/10.1080/07370008.2013.799168>

- Bair, M. A., & Bair, D. E. (2013). Failure, The Next Generation: Why Rigorous Standards are not Sufficient to Improve Science Learning. *International Journal of Education Policy and Leadership*, 9(5). <https://doi.org/10.22230/ijepl.2014v9n5a562>
- Barker, L. J., Garvin-Doxas, K., & Jackson, M. (2002). Defensive climate in the computer science classroom. *SIGCSE Bull.*, 34(1), 43–47. <https://doi.org/10.1145/563517.563354>
- Barnes, T. M. (2013). Type I: Collaborative Research: FRABJOUS CS - Framing a Rigorous Approach to Beauty and Joy for Outreach to Underrepresented Students in Computing at Scale [ADS Bibcode: 2013nsf....1346922B]. *NSF Award*, 13, 46922. Retrieved October 17, 2025, from <https://ui.adsabs.harvard.edu/abs/2013nsf....1346922B>
- Batra, R., & Atiq, Z. (2025). Emotions and Self-Efficacy of Undergraduate Computing and Engineering Students: A Systematic Literature Review. *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*, 88–94. <https://doi.org/10.1145/3641554.3701904>
- Baynes, A., & Norris, A. (2021). Professional Development for High School Computer Science Teachers on Data Science through a Feminist Lens. *2021 IEEE Frontiers in Education Conference (FIE)*, 1–5. <https://doi.org/10.1109/FIE49875.2021.9637249>
- Bean, N. H., Weese, J. L., Feldhausen, R., Allen, D. S., & Friend, M. (2025). Building the Cyber Pipeline: Providing CS Education For Rural K-12 Schools. *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*, 102–108. <https://doi.org/10.1145/3641554.3701840>
- Beddoes, K. (2011). Problematizations of women's underrepresentation: Comparing educator interviews with the literature. *2011 Frontiers in Education Conference (FIE)*. <https://doi.org/10.1109/FIE.2011.6143035>
- Berglund, A., Daniels, M., & Pears, A. (2006). Qualitative research projects in computing education research: An overview. *Proceedings of the 8th Australasian Conference on Computing Education - Volume 52*, 52, 25–33. <https://doi.org/10.5555/1151869.1151875>
- Bettin, B. (2023). Challenges, Choice, & Change: Experiences and Reflections from the First Semester of a Technology and Human Futures Course. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, 235–241. <https://doi.org/10.1145/3545945.3569872>
- Birt, L., Scott, S., Cavers, D., Campbell, C., & Walter, F. (2016). Member Checking. *Qualitative Health Research*. <https://doi.org/10.1177/1049732316654870>
- Blackmore, J. (2019). Feminism and neo/liberalism: Contesting education's possibilities [eprint: <https://doi.org/10.1080/01596306.2019.1569877>]. *Discourse: Studies in the Cultural Politics of Education*, 40(2), 176–190. <https://doi.org/10.1080/01596306.2019.1569877>

- Blikstein, P. (2012). Bifocal modeling: A study on the learning outcomes of comparing physical and computational models linked in real time. *Proceedings of the 14th ACM international conference on Multimodal interaction*, 257–264. <https://doi.org/10.1145/2388676.2388729>
- Boda, P. A., & Brown, B. (2020). Designing for Relationality in Virtual Reality: Context-Specific Learning as a Primer for Content Relevancy. *Journal of Science Education and Technology*, 29(5), 691–702. <https://doi.org/10.1007/s10956-020-09849-1>
- Botes, M., & Rossi, A. (2024). At the Verge of 'Is' and 'Could Be': Storytelling as Medium to Develop Critical Ethical Skills. In E. Hildt, K. Laas, E. M. Brey, & C. Z. Miller (Eds.), *Building Inclusive Ethical Cultures in STEM* (pp. 313–330). Springer International Publishing. https://doi.org/10.1007/978-3-031-51560-6_17
- Braswell, K. M., Johnson, J., Brown, B., & Payton, J. (2021). Pivoting During a Pandemic: Designing a Virtual Summer Camp to Increase Confidence of Black and Latina girls. *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, 686–691. <https://doi.org/10.1145/3408877.3432380>
- Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), 77–101. <https://doi.org/10.1191/1478088706qp063oa>
- Braun, V., & Clarke, V. (2012). Thematic analysis. In *APA handbook of research methods in psychology, Vol 2: Research designs: Quantitative, qualitative, neuropsychological, and biological* (pp. 57–71). American Psychological Association. <https://doi.org/10.1037/13620-004>
- Braun, V., & Clarke, V. (2021). Can I use TA? Should I use TA? Should I *not* use TA? Comparing reflexive thematic analysis and other pattern-based qualitative analytic approaches. *Counselling and Psychotherapy Research*, 21(1), 37–47. <https://doi.org/10.1002/capr.12360>
- Brown, N. C. C., Hermans, F. F. J., & Margulieux, L. E. (2023). 10 Things Software Developers Should Learn about Learning. *Communications of the ACM*, 67(1), 78–87. <https://doi.org/10.1145/3584859>
- Buechley, L., Elumeze, N., Dodson, C., & Eisenberg, M. (2005). Quilt Snaps: A Fabric Based Computational Construction Kit. *IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'05)*, 219–221. <https://doi.org/10.1109/WMTE.2005.55>
- Buechley, L., Eisenberg, M., Catchen, J., & Crockett, A. (2008). The LilyPad Arduino: Using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 423–432. <https://doi.org/10.1145/1357054.1357123>
- Camfield, E. K. (2025, February). *Joy-Centered Pedagogy in Higher Education: Uplifting Teaching and Learning for All* [Google-Books-ID: 7RVEEQAAQBAJ]. Taylor & Francis.

- Campbell, C. M., & Dortch, D. (2018). Reconsidering Academic Rigor: Posing and Supporting Rigorous Course Practices at Two Research Institutions. *Teachers College Record*, 120(5), 1–42. <https://doi.org/10.1177/016146811812000503>
- Carrigan, C. (2024). *Cracking the bro code*. The MIT press. <https://doi.org/https://doi.org/10.7551/mitpress/14883.001.0001>
- Carruthers, S. (2024). Indigenization and Decolonization of CS Education 2010-2023: A Bibliometric Analysis. *Proceedings of the 26th Western Canadian Conference on Computing Education*, 1–8. <https://doi.org/10.1145/3660650.3660658>
- Chang, M. A., Wong, R. Y., Breideband, T., Philip, T. M., McKoy, A., Cortez, A., & D'Mello, S. K. (2024). Co-design Partners as Transformative Learners: Imagining Ideal Technology for Schools by Centering Speculative Relationships. *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 1–15. <https://doi.org/10.1145/3613904.3642559>
- Charleston, L. J., & Charleston, S. A. (2014). Using Culturally Responsive Practices to Broaden Participation in the Educational Pipeline: Addressing the Unfinished Business of Brown in the Field of Computing Sciences. *Journal of Negro Education*, 83(3), 400–419. Retrieved March 10, 2024, from <https://muse.jhu.edu/pub/417/article/802984>
- Chávez, V. C. (2023). How My Students and I (Re)Discovered the Joy of Computing in CS2. *ACM Inroads*, 14(2), 36–39. <https://doi.org/10.1145/3596918>
- Cheryan, S., & Markus, H. R. (2020). Masculine defaults: Identifying and mitigating hidden cultural biases. *Psychological Review*, 127(6), 1022–1052. <https://doi.org/10.1037/rev0000209>
- Cheryan, S., Master, A., & Meltzoff, A. (2023). There Are Too Few Women in Computer Science and Engineering. *Scientific American*. Retrieved February 17, 2025, from <https://www.scientificamerican.com/article/there-are-too-few-women-in-computer-science-and-engineering/>
- Childs, J., Zarch, R., Dunton, S., Madkins, T. c., Trautmann, K., Taylor, Z. W., Jacobson, M., Ozturk, S., & Ottenbreit-Leftwich, A. (2024). Advancing Equity and Access: Addressing the Side Effects of Broadening Participation in Computer Science K–12 Education. *Review of Research in Education*, 48(1), 121–153. <https://doi.org/10.3102/0091732X241286475>
- Cicuto, E., Gennari, R., & Meonio, A. (2024). Co-design and Physical Computing with ADHD Learners: Preliminary Investigations. In A. Bramwell-Dicks, A. Evans, M. Winckler, H. Petrie, & J. Abdelnour-Nocera (Eds.), *Design for Equality and Justice* (pp. 124–131). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-61688-4_11
- Clarke, V., & Braun, V. (2017). Thematic analysis. *The Journal of Positive Psychology*, 12(3), 297–298. <https://doi.org/10.1080/17439760.2016.1262613>

- Code.org. (2022). 2022 State of CS Report [Publication Title: Code.org]. Retrieved August 4, 2023, from <https://advocacy.code.org/stateofcs>
- Coenraad, M., Palmer, J., Franklin, D., & Weintrop, D. (2019). Enacting Identities: Participatory Design as a Context for Youth to Reflect, Project, and Apply their Emerging Identities. *Proceedings of the 18th ACM International Conference on Interaction Design and Children*, 185–196. <https://doi.org/10.1145/3311927.3323148>
- Computer Science Pedagogical Content Knowledge: Characterizing Teacher Performance: ACM Transactions on Computing Education: Vol 19, No 3. (n.d.). Retrieved March 1, 2025, from <https://dl.acm.org/doi/abs/10.1145/3303770>
- Convertino, C. (2020). Nuancing the discourse of underrepresentation: A feminist post-structural analysis of gender inequality in computer science education in the US. *Gender and Education*, 32(5), 594–607. <https://doi.org/10.1080/09540253.2019.1632417>
- Cooke, L., Vogel, S., Lachney, M., & Santo, R. (2019). Culturally Responsive Computing: Supporting Diverse Justice Projects In/As Computer Science Education. *2019 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, 1–2. <https://doi.org/10.1109/RESPECT46404.2019.8985928>
- Crawford-Garrett, K., & Carbajal, D. R. (2023). *Promising Pedagogies for Teacher Inquiry and Practice: Teaching Out Loud* [Google-Books-ID: XPKnEAAAQBAJ]. Teachers College Press.
- Culver, K. C., Braxton, J. M., & Pascarella, E. T. (2021). What We Talk about When We Talk about Rigor: Examining Conceptions of Academic Rigor. *The Journal of Higher Education*, 92(7), 1140–1163. <https://doi.org/10.1080/00221546.2021.1920825>
- Davis, J., Lachney, M., Zatz, Z., Babbitt, W., & Eglash, R. (2019). A Cultural Computing Curriculum. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 1171–1175. <https://doi.org/10.1145/3287324.3287439>
- Davis, J. D., McDuffie, A. R., Drake, C., & Seiwel, A. L. (2019). Teachers' perceptions of the official curriculum: Problem solving and rigor. *International Journal of Educational Research*, 93, 91–100. <https://doi.org/10.1016/j.ijer.2018.10.002>
- Davis, K., White, S. V., Becton-Consuegra, D., & Scott, A. (2021, June). *Culturally Responsive-Sustaining Computer Science Education: A Framework* (tech. rep.). Kapor Center. <https://www.kaporcenter.org/culturally-responsive-sustaining-computer-science-education-a-framework/>
- de Wit, S., Hermans, F., Specht, M., & Aivaloglou, E. (2023). Children's Interest in a CS Career: Exploring Age, Gender, Computer Interests, Programming Experience and Stereotypes. *Proceedings of the 2023 ACM*

- Conference on International Computing Education Research - Volume 1, 1*, 245–255. <https://doi.org/10.1145/3568813.3600131>
- Delcourt, C., Venkatagiri, S., & Chandrasekharan, E. (2024). What's in a Social Computing Course: Analyzing Computer and Information Science Syllabi. *Proceedings of the 6th Annual Symposium on HCI Education*, 1–8. <https://doi.org/10.1145/3658619.3658623>
- DeLyser, L. A., Wortel-London, S., Wright, L., & Bora, A. (2019). Understanding our Human Resources: District Leadership Efforts at Understanding CS Education Implementation in Their Own Buildings. *Proceedings of the 2019 ACM Conference on International Computing Education Research*, 297. <https://doi.org/10.1145/3291279.3341206>
- Design-Based Research: An Emerging Paradigm for Educational Inquiry. (2003). *Educational Researcher*, 32(1), 5–8. <https://doi.org/10.3102/0013189X032001005>
- Desportes, K., McDermott, K., Bergner, Y., & Payne, W. (2022). “Go[ing] Hard...as a Woman of Color”: A Case Study Examining Identity Work within a Performative Dance and Computing Learning Environment. *ACM Transactions on Computing Education*, 22(4), 49:1–49:29. <https://doi.org/10.1145/3531000>
- Dijkstra, E. W. (1988, December). On the cruelty of really teaching computing science.
- DiSalvo, B., & DiSalvo, C. (2014). Designing for Democracy in Education: Participatory Design and the Learning Sciences. *ICLS*, 792–799. Retrieved March 2, 2025, from <https://repository.isls.org//handle/1/1196>
- DiSalvo, B. J., Yardi, S., Guzdial, M., McKlin, T., Meadows, C., Perry, K., & Bruckman, A. (2011). African American men constructing computing identity. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2967–2970. <https://doi.org/10.1145/1978942.1979381>
- Domina, T., Pharris-Ciurej, N., Penner, A. M., Penner, E. K., Brummet, Q., Porter, S. R., & Sanabria, T. (2018). Is Free and Reduced-Price Lunch a Valid Measure of Educational Disadvantage? *Educational Researcher*, 47(9), 539–555. <https://doi.org/10.3102/0013189X18797609>
- Dong, Y., Shabrina, P., Marwan, S., & Barnes, T. (2021). You Really Need Help: Exploring Expert Reasons for Intervention During Block-based Programming Assignments. *Proceedings of the 17th ACM Conference on International Computing Education Research*, 334–346. <https://doi.org/10.1145/3446871.3469764>
- Dorodchi, M., & Dehbozorgi, N. (2017). Addressing the Paradox of Fun and Rigor in Learning Programming. *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education*, 370–370. <https://doi.org/10.1145/3059009.3073004>
- Dougherty, J. P., & Wonnacott, D. G. (2005). Use and assessment of a rigorous approach to CS1. *ACM SIGCSE Bulletin*, 37(1), 251–255. <https://doi.org/10.1145/1047344.1047431>

- Downes, T., & Looker, D. (2011). Factors that influence students' plans to take computing and information technology subjects in senior secondary school. *Computer Science Education*, 21(2), 175–199. <https://doi.org/10.1080/08993408.2011.579811>
- Draeger, J., del Prado Hill, P., Hunter, L. R., & Mahler, R. (2013). The Anatomy of Academic Rigor: The Story of One Institutional Journey. *Innovative Higher Education*, 38(4), 267–279. <https://doi.org/10.1007/s10755-012-9246-8>
- Draeger, J., del Prado Hill, P., & Mahler, R. (2015). Developing a Student Conception of Academic Rigor. *Innovative Higher Education*, 40(3), 215–228. <https://doi.org/10.1007/s10755-014-9308-1>
- Druin, A. (2002). The role of children in the design of new technology [_eprint: <https://doi.org/10.1080/01449290110108659>]. *Behaviour & Information Technology*, 21(1), 1–25. <https://doi.org/10.1080/01449290110108659>
- Dunne, A. (2013). *Speculative everything: Design, fiction, and social dreaming*. The MIT Press.
- Eglash, R., Gilbert, J. E., Taylor, V., & Geier, S. R. (2013). Culturally Responsive Computing in Urban, After-School Contexts: Two Approaches. *Urban Education*, 48(5), 629–656. <https://doi.org/10.1177/0042085913499211>
- Emiliussen, J., Engelsen, S., Christiansen, R., & Klausen, S. H. (2021). We are all in it!: Phenomenological Qualitative Research and Embeddedness. *International Journal of Qualitative Methods*, 20. <https://doi.org/10.1177/1609406921995304>
- Ensmenger, N. (2015). “Beards, Sandals, and Other Signs of Rugged Individualism”: Masculine Culture within the Computing Professions. *Osiris*, 30(1), 38–65. <https://doi.org/10.1086/682955>
- Erete, S., Thomas, K., Nacu, D., Dickinson, J., Thompson, N., & Pinkard, N. (2021). Applying a Transformative Justice Approach to Encourage the Participation of Black and Latina Girls in Computing. *ACM Trans. Comput. Educ.*, 21(4), 1–24. <https://doi.org/10.1145/3451345>
- Ericson, B., Guzdial, M., & Biggers, M. (2007). Improving secondary CS education. *Proceedings of the 38th SIGCSE technical symposium on Computer science education - SIGCSE '07*, 298–301. <https://doi.org/10.1145/1227310.1227416>
- Eschenbach, E., Cashman, E., Waller, A., & Lord, S. (2005). Incorporating feminist pedagogy into the engineering learning experience. *Proceedings Frontiers in Education 35th Annual Conference*, F4H–8. <https://doi.org/10.1109/FIE.2005.1612156>
- Everson, J., Kivuva, F. M., & Ko, A. J. (2022). "A Key to Reducing Inequities in Like, AI, is by Reducing Inequities Everywhere First": Emerging Critical Consciousness in a Co-Constructed Secondary CS Classroom. *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1, 1*, 209–215. <https://doi.org/10.1145/3478431.3499395>

- Everson, J., Kivuva, F. M., Montes De Haro, C., & Ko, A. J. (2024). Cultural-Centric Computational Embroidery. *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, 673–679. <https://doi.org/10.1145/3626252.3630818>
- Everson, J., Kivuva, F. M., Montes De Haro, C., & Ko, A. J. (2026). Culture-centric computational embroidery as a medium for learning computer science [_eprint: <https://doi.org/10.1080/08993408.2026.2618836>]. *Computer Science Education*, 0(0), 1–34. <https://doi.org/10.1080/08993408.2026.2618836>
- Everson, J., Kivuva, F. M., Sherif, E., Oleson, A., & Ko, A. J. (2026). Deconstructing Conceptions of Rigor in Computer Science Education. *ACM Trans. Comput. Educ.*, 26(1), 16:1–16:25. <https://doi.org/10.1145/3776542>
- Everson, J., Landesman, R., Kivuva, F. M., & Ko, A. J. (2025). Dreaming of Difference: Imagining the Future of CS Education Pedagogy with Secondary Students. *Proceedings of the 2025 ACM Conference on International Computing Education Research V.1*, 394–406. <https://doi.org/10.1145/3702652.3744203>
- Federici, S., & Stern, L. (2011). A Constructionist Approach to Computer Science, 1352–1361.
- Fey, J., Dagan, E., Márquez Segura, E., & Isbister, K. (2022). Anywear Academy: A Larp-based Camp to Inspire Computational Interest in Middle School Girls. *Designing Interactive Systems Conference*, 1192–1208. <https://doi.org/10.1145/3532106.3533532>
- Fields, D. A., Kafai, Y., Nakajima, T., Goode, J., & Margolis, J. (2018). Putting Making into High School Computer Science Classrooms: Promoting Equity in Teaching and Learning with Electronic Textiles in Exploring Computer Science [_eprint: <https://doi.org/10.1080/10665684.2018.1436998>]. *Equity & Excellence in Education*, 51(1), 21–35. <https://doi.org/10.1080/10665684.2018.1436998>
- Fields, D. A. (2009). What do Students Gain from a Week at Science Camp? Youth perceptions and the design of an immersive, research-oriented astronomy camp [_eprint: <https://doi.org/10.1080/09500690701648291>]. *International Journal of Science Education*, 31(2), 151–171. <https://doi.org/10.1080/09500690701648291>
- Fiesler, C. (2021). Ethical Speculation in the Computing Classroom. *2021 Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. <https://doi.org/10.1109/RESPECT51740.2021.9620641>
- Flagg Young, E. (1900). *Isolation in the School* [Doctoral dissertation, The University of Chicago]. <https://doi.org/10.1086/678221>
- Fletcher, C. L., & Warner, J. R. (2021). CAPE: A framework for assessing equity throughout the computer science education ecosystem. *Communications of the ACM*, 64(2), 23–25. <https://doi.org/10.1145/3442373>
- for Science, N. C., & (NCSES), E. S. (2023). *Diversity and STEM: Women, Minorities, and Persons with Disabilities 2023* (Special Report NSF 23-315). Alexandria, VA: National Science Foundation. Retrieved February 21, 2025, from <https://ncses.nsf.gov/wmpd>

- Freire, P. (2017). *Pedagogy of the oppressed*. Penguin Classics.
- Frieze, C., & Quesenberry, J. L. (2013). From difference to diversity: Including women in the changing face of computing. *Proceeding of the 44th ACM technical symposium on Computer science education*, 445–450. <https://doi.org/10.1145/2445196.2445327>
- Gannon, K. (2023, May). Advice | Why Calls for a ‘Return to Rigor’ Are Wrong [Section: Advice]. Retrieved April 29, 2024, from <https://www.chronicle.com/article/why-calls-for-a-return-to-rigor-are-wrong>
- Garcia, D. D., Cutler, R., Dodds, Z., Roberts, E., & Young, A. (2009). Rediscovering the passion, beauty, joy, and awe: Making computing fun again, continued. *Proceedings of the 40th ACM technical symposium on Computer science education*, 65–66. <https://doi.org/10.1145/1508865.1508889>
- Garcia, D. D., & Harvey, B. (2011). Collaborative Research: Type 1: FRABJIOUS CS - Framing a Rigorous Approach to Beauty and Joy for Outreach to Underrepresented Students in Computing at Scale [ADS Bibcode: 2011nsf....1138596G]. *NSF Award*, 11, 38596. Retrieved October 17, 2025, from <https://ui.adsabs.harvard.edu/abs/2011nsf....1138596G>
- Garr-Schultz, A., Muragishi, G. A., Mortejo, T. A., & Cheryan, S. (2023). Masculine Defaults in Academic Science, Technology, Engineering, and Mathematics (STEM) Fields. *Psychological Science in the Public Interest*, 24(1), 1–9. <https://doi.org/10.1177/15291006231170829>
- Gaskins, N. R. (2021). *Techno-vernacular creativity and innovation: Culturally relevant making inside and outside of the classroom*. MIT Press.
- Gautam, A., Kulkarni, A., Hug, S., Lehr, J., & Yoon, I. (2024). Socially Responsible Computing in an Introductory Course. *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, 373–379. <https://doi.org/10.1145/3626252.3630926>
- Gay, G. (2002). Preparing for Culturally Responsive Teaching. *Journal of Teacher Education*, 53(2), 106–116. <https://doi.org/10.1177/0022487102053002003>
- Gay, G. (2018, January). *Culturally Responsive Teaching: Theory, Research, and Practice. Third Edition. Multicultural Education Series*. Teachers College Press.
- Ginat, D. (2007a). Elaborating heuristic reasoning and rigor with mathematical games. *ACM SIGCSE Bulletin*, 39(4), 32–36. <https://doi.org/10.1145/1345375.1345409>
- Ginat, D. (2007b). Hasty design, futile patching and the elaboration of rigor. *Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education - ITiCSE '07*, 161–165. <https://doi.org/10.1145/1268784.1268832>
- Greening, T. (1999). Gender stereotyping in a computer science course. *The proceedings of the thirtieth SIGCSE technical symposium on Computer science education*, 203–207. <https://doi.org/10.1145/299649.299756>

- Guridi, S., Vicencio, T., & Gajardo, R. (2021). Arpilleras Parlantes: Designing Educational Material for the Creation of Interactive Textile Art Based on a Traditional Chilean Craft. *Proceedings of the Fifteenth International Conference on Tangible, Embedded, and Embodied Interaction*, 1–11. <https://doi.org/10.1145/3430524.3440657>
- Guzdial, M., Ericson, B., Mcklin, T., & Engelman, S. (2014). Georgia Computes! An Intervention in a US State, with Formal and Informal Education in a Policy Context. *ACM Transactions on Computing Education*, 14(2), 1–29. <https://doi.org/10.1145/2602488>
- Hall, A., & Steele, B. (2024). Joy as subversive defiance. *Journal of International Political Theory*, 20(3), 232–246. <https://doi.org/10.1177/17550882241284043>
- Hammer, D., & Berland, L. K. (2014). Confusing claims for data: A critique of common practices for presenting qualitative research on learning. *Journal of the Learning Sciences*, 23(1), 37–46. <https://doi.org/10.1080/10508406.2013.802652>
- Happe, L., Buhnova, B., Koziolok, A., & Wagner, I. (2021). Effective measures to foster girls' interest in secondary computer science education. *Education and Information Technologies*, 26(3), 2811–2829. <https://doi.org/10.1007/s10639-020-10379-x>
- Harding, S. (1992). Rethinking Standpoint Epistemology: What Is "Strong Objectivity?" *The Centennial Review*, 36(3), 437–470. <https://www.jstor.org/stable/23739232>
- Harvey, B. (2012). *The Beauty and Joy of Computing: Computer Science for Everyone*.
- Hazzan, O., Gal-Ezer, J., & Blum, L. (2008). A model for high school computer science education: The four key elements that make it! *ACM SIGCSE Bulletin*, 40(1), 281–285. <https://doi.org/10.1145/1352135.1352233>
- hooks, b. (1994). *Teaching to transgress: Education as the practice of freedom*. Routledge.
- hooks, b. (2015). *Feminism is for everybody: Passionate politics* (second edition). Routledge. <https://doi.org/10.4324/9781315743189>
- hooks, b. (2018). *All about love: New visions* (First William Morrow paperback edition.). William Morrow.
- Horn, M. S., Banerjee, A., West, M., Pinkard, N., Pratt, A., Freeman, J., Magerko, B., & McKlin, T. (2020). TunePad: Engaging Learners at the Intersection of Music and Code. *ICLS 2020 Proceedings*, 1237–1244.
- Hubwieser, P., Armoni, M., & Giannakos, M. N. (2015). How to Implement Rigorous Computer Science Education in K-12 Schools? Some Answers and Many Questions. *ACM Transactions on Computing Education (TOCE)*, 15(2), 1–12. <https://doi.org/10.1145/2729983>
- Hubwieser, P., Armoni, M., Giannakos, M. N., & Mittermeir, R. T. (2014). Perspectives and Visions of Computer Science Education in Primary and Secondary (K-12) Schools. *ACM Transactions on Computing Education (TOCE)*, 14(2), 7. <https://doi.org/10.1145/2602482>

- Hwang, Y., Das, A., Waite, J., & Sentance, S. (2023). Using a Sociological Lens to Investigate Computing Teachers' Culturally Responsive Classroom Practices. *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1, 1*, 206–221. <https://doi.org/10.1145/3568813.3600112>
- Iriarte, M. P., Koren, N., Blinder, E. B., & Bonsignore, E. (2023). Funds of Identity in Co-Design. *Proceedings of the 22nd Annual ACM Interaction Design and Children Conference*, 568–573. <https://doi.org/10.1145/3585088.3593901>
- Ivey, A., Johnson, S. R., Skorodinsky, M., Snyder, J., & Goode, J. (2021). Abolitionist Computer Science Teaching: Moving from Access to Justice. *2021 Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, 1–4. <https://doi.org/10.1109/RESPECT51740.2021.9620652>
- Jack, J., & Sathy, V. (2021). Advice | It's Time to Cancel the Word 'Rigor'. *The Chronicle of Higher Education*. Retrieved August 26, 2023, from <https://www.chronicle.com/article/its-time-to-cancel-the-word-rigor>
- Jane Margolis, Rachel Estrella, Joanna Goode, Jennifer Jellison Holme, & Kim Nao. (2017). *Stuck in the Shallow End: Education, Race, and Computing*. The MIT Press.
- Jasinski, I., & Lewis, T. E. (2016). Another Kind of Love in Education: Whatever Love. *Interchange*, 47(4), 429–441. <https://doi.org/10.1007/s10780-016-9282-x>
- Jayathirtha, G., Fields, D., Kafai, Y. B., & Chipps, J. (2020). Supporting making online: The role of artifact, teacher and peer interactions in crafting electronic textiles. *Information and Learning Sciences*, 121(5/6), 381–390. <https://doi.org/10.1108/ILS-04-2020-0111>
- Jayathirtha, G., & Kafai, Y. B. (2019). Electronic Textiles in Computer Science Education: A Synthesis of Efforts to Broaden Participation, Increase Interest, and Deepen Learning. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 713–719. <https://doi.org/10.1145/3287324.3287343>
- Johnson, J. E., Jones, J. A., Weidner, T. G., & Manwell, A. K. (2019). Evaluating Academic Rigor, Part II: An Investigation of Student Ratings, Course Grades, and Course Level. *Journal of Assessment and Institutional Effectiveness*, 9(1-2), 49–78. <https://doi.org/10.5325/jasseinsteffe.9.1-2.0049>
- Johnson, J. E., Weidner, T. G., Jones, J. A., & Manwell, A. K. (2018). Evaluating Academic Course Rigor, Part 1: Defining a Nebulous Construct. *Journal of Assessment and Institutional Effectiveness*, 8(1-2), 86–121. <https://doi.org/10.5325/jasseinsteffe.8.1-2.0086>
- Johnson, M. J., Castro, F. E. V., DiSalvo, B., & DesPortes, K. (2023). Chronicles of Exploration: Examining the Materiality of Computational Artifacts. *Proceedings of the 2023 ACM Conference on International Computing Education Research, 1*, 29–47. <https://doi.org/10.1145/3568813.3600132>

- Johnson, S. L., Bishop, J. P., & Rogers, K. D. (2025). Shifting pedagogically: Incorporating the social, cultural, and emotional dimensions of student learning to develop STEM-identities in computer science. *Journal of Applied Developmental Psychology*, 97. <https://doi.org/10.1016/j.appdev.2025.101757>
- Jones, S. T., & Melo, N. (2020). 'Anti-blackness is no glitch': The need for critical conversations within computer science education. *XRDS*, 27(2), 42–46. <https://doi.org/10.1145/3433134>
- Kafai, Y., Fields, D., & Searle, K. (2014). Electronic Textiles as Disruptive Designs: Supporting and Challenging Maker Activities in Schools. *Harvard Educational Review*, 84(4), 532–556. <https://doi.org/10.17763/haer.84.4.46m7372370214783>
- Kafai, Y., Searle, K., Martinez, C., & Brayboy, B. (2014). Ethnocomputing with electronic textiles: Culturally responsive open design to broaden participation in computing in American indian youth and communities. *Proceedings of the 45th ACM technical symposium on Computer science education*, 241–246. <https://doi.org/10.1145/2538862.2538903>
- Kafai, Y. B., Fields, D. A., Lui, D. A., Walker, J. T., Shaw, M. S., Jayathirtha, G., Nakajima, T. M., Goode, J., & Giang, M. T. (2019). Stitching the Loop with Electronic Textiles: Promoting Equity in High School Students' Competencies and Perceptions of Computer Science. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 1176–1182. <https://doi.org/10.1145/3287324.3287426>
- Kareta, M., Mann, S., Guruge, D. D., Licorish, S., & Clear, A. (2023). Decolonising Computer Science Education - A Global Perspective. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, 1097–1102. <https://doi.org/10.1145/3545945.3569870>
- Karlin, M., Stephany, C., & Reed, M. (2023). Coding to Connect: Centering Joy and Community in Elementary Computer Science Education. In *Dialogue/En Diálogo*, 1(1), 71–84. <https://doi.org/10.46787/dialogue.v1i1.3327>
- Kawas, S., Yuan, Y., DeWitt, A., Jin, Q., Kirchner, S., Bilger, A., Grantham, E., Kientz, J. A., Tartaro, A., & Yarosh, S. (2020). Another decade of IDC research: Examining and reflecting on values and ethics. *Proceedings of the Interaction Design and Children Conference*, 205–215. <https://doi.org/10.1145/3392063.3394436>
- Keune, A. (2022a). Performing Algorithms: Weaving as Promising Context for Computational Learning. Retrieved September 12, 2025, from <https://repository.isls.org/handle/1/8904>
- Keune, A. (2022b). Material syntonicity: Examining computational performance and its materiality through weaving and sewing crafts [eprint: <https://doi.org/10.1080/10508406.2022.2100704>]. *Journal of the Learning Sciences*, 31(4-5), 477–508. <https://doi.org/10.1080/10508406.2022.2100704>

- Keune, A. (2024). Learning within fiber-crafted algorithms: Posthumanist perspectives for capturing human-material collaboration. *International Journal of Computer-Supported Collaborative Learning*, 19(1), 37–65. <https://doi.org/10.1007/s11412-023-09412-1>
- Khan, A. H., Ejaz, N., Matthews, S., Snow, S., & Matthews, B. (2021). Speculative Design for Education: Using Participatory Methods to Map Design Challenges and Opportunities in Pakistan. *Designing Interactive Systems Conference 2021*, 1748–1764. <https://doi.org/10.1145/3461778.3462117>
- Kimmerer, R. W. (2020). *Braiding sweetgrass: Indigenous wisdom, scientific knowledge and the teachings of plants*. Penguin Books.
- Kirdani-Ryan, M., & Ko, A. J. (2024). Neurodivergent Legitimacy in Computing Spaces. *ACM Trans. Comput. Educ.*, 24(4), 49:1–49:28. <https://doi.org/10.1145/3690651>
- Klassen, S., & Fiesler, C. (2022). "Run Wild a Little With Your Imagination": Ethical Speculation in Computing Education with Black Mirror. *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education*, 1, 836–842. <https://doi.org/10.1145/3478431.3499308>
- Klassen, S., & Fiesler, C. (2023). The Stoop: Speculation on Positive Futures of Black Digital Spaces. *Proceedings of the ACM on Human-Computer Interaction*, 7(GROUP), 1–24. <https://doi.org/10.1145/3567567>
- Ko, A. J., Beitlers, A., Everson, J., Wortzman, B., & Gallagher, D. (2023). Proposing, Planning, and Teaching an Equity- and Justice-Centered Secondary Pre-Service CS Teacher Education Program. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education*, 583–589. <https://doi.org/10.1145/3545945.3569735>
- Kumar, S., & Kremer-Herman, N. (2019). Integrating Ethics Across Computing: An Experience Report of Three Computing Courses Engaging Ethics and Societal Impact through Roleplaying, Case Studies, and Service Learning. *2019 IEEE Frontiers in Education Conference (FIE)*, 1–5. <https://doi.org/10.1109/FIE43999.2019.9028568>
- Lachney, M., Bennett, A. G., Eglash, R., Yadav, A., & Moudgalya, S. (2021). Teaching in an open village: A case study on culturally responsive computing in compulsory education. *Computer Science Education*, 31(4), 462–488. <https://doi.org/10.1080/08993408.2021.1874228>
- Ladner, R. E., & Stefik, A. (2017). AccessCSforall: Making computer science accessible to K-12 students in the United States. *ACM SIGACCESS Accessibility and Computing*, (118), 3–8. <https://doi.org/10.1145/3124144.3124145>
- Ladson-Billings, G. (1995). But that's just good teaching! The case for culturally relevant pedagogy. *Theory Into Practice*, 34(3), 159–165. <https://doi.org/10.1080/00405849509543675>
- Ladson-Billings, G. (1995). Toward a Theory of Culturally Relevant Pedagogy. *American Educational Research Journal*, 32(3), 465–491. <https://doi.org/10.3102/00028312032003465>

- Ladson-Billings, G. (2021). Three Decades of Culturally Relevant, Responsive, & Sustaining Pedagogy: What Lies Ahead? *The Educational Forum*, 85(4), 351–354. <https://doi.org/10.1080/00131725.2021.1957632>
- Leddy, S. (2024). *Teaching where you are: Weaving Indigenous and slow principles and pedagogies*. University of Toronto Press.
- Lee, H. H., Celedón-Pattichis, S., Pattichis, M. S., Johnson, A. R., Cantú, E., Tovar, I., & López-Leiva, C. A. (2023). Knowing and Enjoying: Expanding Latinx Students' Experiences With an Integrated Computer Science and Mathematics Curriculum. <https://doi.org/10.3102/2012330>
- Lee, Y.-C., & Albaugh, L. (2021). Hybrid Embroidery Games: Playing with Materials, Machines, and People. *Designing Interactive Systems Conference 2021*, 749–762. <https://doi.org/10.1145/3461778.3462019>
- Lewis, C. M. (2023). Examples of Unsuccessful Use of Code Comprehension Strategies: A Resource for Developing Code Comprehension Pedagogy. *Proceedings of the 2023 ACM Conference on International Computing Education Research V.1*, 15–28. <https://doi.org/10.1145/3568813.3600116>
- 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. *Proceedings of the 2016 ACM Conference on International Computing Education Research*, 23–32. <https://doi.org/10.1145/2960310.2960332>
- Lewis Ellison, T. (2017). Digital Participation, Agency, and Choice: An African American Youth's Digital Storytelling About Minecraft. *Journal of Adolescent & Adult Literacy*, 61(1), 25–35. <https://doi.org/10.1002/jaal.645>
- Li, Y., Nwogu, J., Buddemeyer, A., Solyst, J., Lee, J., Walker, E., Ogan, A., & Stewart, A. E. (2023). "I Want to Be Unique From Other Robots": Positioning Girls as Co-creators of Social Robots in Culturally-Responsive Computing Education. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 1–14. <https://doi.org/10.1145/3544548.3581272>
- Linn, M. C. (2009, August). *The Computer Clubhouse: Constructionism and Creativity in Youth Communities* (Y. B. Kafai, K. A. Peppler, & R. N. Chapman, Eds.). Teachers College Press.
- Lord, S., Cashman, E., Eschenbach, E., & Waller, A. (2005). Feminism and engineering. *Proceedings Frontiers in Education 35th Annual Conference*, F4H–14. <https://doi.org/10.1109/FIE.2005.1612157>
- Luciano, A. P. G., Fusinato, P. A., Gomes, L. C., Luciano, A., & Takai, H. (2019). The educational robotics and Arduino platform: Constructionist learning strategies to the teaching of physics. *Journal of Physics: Conference Series*, 1286(1), 012044. <https://doi.org/10.1088/1742-6596/1286/1/012044>
- Lyon, L. A., & Clayton, C. (2021). Arising of Informal Women's Learn-to-code Communities: Activity Systems as Incubators. *ACM Trans. Comput. Educ.*, 21(2), 12:1–12:24. <https://doi.org/10.1145/3433167>

- Malinverni, L., Schaper, M.-M., Rubegni, E., & Tamashiro, M. A. (2025). Scaffolding Children's critical reflection on intelligent technologies: Opportunities from speculative fiction. *International Journal of Child-Computer Interaction*, 43, 100727. <https://doi.org/10.1016/j.ijcci.2025.100727>
- Margolis, J., & Fisher, A. (2002). *Unlocking the Clubhouse: Women in Computing* [Google-Books-ID: StwGQw45YoEC]. MIT Press.
- Margolis, J., Ryoo, J. J., Sandoval, C. D. M., Lee, C., Goode, J., & Chapman, G. (2012). Beyond access: Broadening participation in high school computer science. *ACM Inroads*, 3(4), 72–78. <https://doi.org/10.1145/2381083.2381102>
- Margulieux, L., Denny, P., Cunningham, K., Deutsch, M., & Shapiro, B. R. (2021). When Wrong is Right: The Instructional Power of Multiple Conceptions. *Proceedings of the 17th ACM Conference on International Computing Education Research*, 184–197. <https://doi.org/10.1145/3446871.3469750>
- Mark, J., & Klein, K. (2019). Beauty and Joy of Computing: 2016-17 Findings from an AP CS Principles course. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 627–633. <https://doi.org/10.1145/3287324.3287375>
- Mark, J., Zeringue, J. K., Klein, K., Mitchell, T., & Olivares, J. A. (2021). Why Do Students Enroll in AP CSP? *2021 Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, 1–9. <https://doi.org/10.1109/RESPECT51740.2021.9620546>
- Mayhew, E. J., & Patitsas, E. (2023). Critical Pedagogy in Practice in the Computing Classroom. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, 1076–1082. <https://doi.org/10.1145/3545945.3569840>
- McDonald, N., Akinsiku, A., Hunter-Cevera, J., Sanchez, M., Kephart, K., Berczynski, M., & Mentis, H. M. (2022). Responsible Computing: A Longitudinal Study of a Peer-led Ethics Learning Framework. *ACM Trans. Comput. Educ.*, 22(4), 47:1–47:21. <https://doi.org/10.1145/3469130>
- McDowell, C., Werner, L., Bullock, H. E., & Fernald, J. (2006). Pair programming improves student retention, confidence, and program quality. *Commun. ACM*, 49(8), 90–95. <https://doi.org/10.1145/1145287.1145293>
- McGill, M. M., Thompson, A., Gransbury, I., Heckman, S., Rosato, J., & Delyser, L. A. (2023). Building upon the CAPE Framework for Broader Understanding of Capacity in K-12 CS Education. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, 577–582. <https://doi.org/10.1145/3545945.3569799>
- McNutt, A. M., Cohen, S., & Chugh, R. (2025). Slowness, Politics, and Joy: Values That Guide Technology Choices in Creative Coding Classrooms. *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, 1–16. <https://doi.org/10.1145/3706598.3713472>

- Meza, F., Torres, N., & Vásquez, A. (2020). Fostering Engagement and Creativity through Programming: The Beauty and Joy of Computing in a First-year Engineering class. *2020 39th International Conference of the Chilean Computer Science Society (SCCC)*, 1–5. <https://doi.org/10.1109/SCCC51225.2020.9281159>
- Mills, J., Bonner, A., & Francis, K. (2006). The Development of Constructivist Grounded Theory. *International Journal of Qualitative Methods*, 5(1), 25–35. <https://doi.org/10.1177/160940690600500103>
- Mishkin, A. (2019). Applying Self-Determination Theory towards Motivating Young Women in Computer Science. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 1025–1031. <https://doi.org/10.1145/3287324.3287389>
- Moll, L. C., Amanti, C., Neff, N., Deborah, & Gonzalez, N. (1992). Funds of knowledge for teaching: Using a qualitative approach to connect homes and classrooms [eprint: <https://doi.org/10.1080/00405849209543534>]. *Theory Into Practice*, 31(2), 132–141. <https://doi.org/10.1080/00405849209543534>
- Mönig, J. (2020). Building Blocks for Powerful Ideas: Designing a Programming Language to Teach the Beauty and Joy of Computing. *Congreso Internacional de Ingeniería de Sistemas*, 45–46. <https://doi.org/10.26439/ciis2019.5499>
- Montgomery, N. (2017). *Joyful militancy: Building resistance in toxic times*. AK Press.
- Morales-Navarro, L., Thompson, N., Kafai, Y., Shaw, M., & Pinkard, N. (2022). Reimagining and Co-designing with Youth an Hour of Code Activity for Critical Engagement with Computing. *Interaction Design and Children*, 288–296. <https://doi.org/10.1145/3501712.3529730>
- Moreno, C., Shalf, M., & Roque, R. (2024). Visualizing the Unseen Design Work of Educators. *Designing Interactive Systems Conference*, 1956–1969. <https://doi.org/10.1145/3643834.3660751>
- Muhammad, G. E. (2022). Cultivating Genius and Joy in Education through Historically Responsive Literacy. *Language Arts*, 99(3), 195–204. <https://doi.org/10.58680/la202231623>
- Nakajima, T. M., & Goode, J. (2019). Teachers' Approaches to Mak(e)ing Computing Culturally Responsive: Electronic-Textiles in Exploring Computer Science Classes. *2019 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, 1–8. <https://doi.org/10.1109/RESPECT46404.2019.8985870>
- Nelson, C. E. (2010). Dysfunctional Illusions of Rigor. *To Improve the Academy*, 28(1), 177–192. <https://doi.org/10.1002/j.2334-4822.2010.tb00602.x>
- Nelson, G. L., Xie, B., & Ko, A. J. (2017). Comprehension First: Evaluating a Novel Pedagogy and Tutoring System for Program Tracing in CS1. *Proceedings of the 2017 ACM Conference on International Computing Education Research*, 2–11. <https://doi.org/10.1145/3105726.3106178>

- Newton, K. J., Leonard, J., Buss, A., Wright, C. G., & Barnes-Johnson, J. (2020). Informal STEM: Learning with robotics and game design in an urban context. *Journal of Research on Technology in Education*, 52(2), 129–147. <https://doi.org/10.1080/15391523.2020.1713263>
- Nias, J., & Holmes, J. (2023). Imagine a Future Rooted in Culture: Exploring Afrofuturism in Computing Education at HBCUs. *2023 Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, 167–171. <https://doi.org/10.1109/RESPECT60069.2023.00040>
- Nolan, K., Faherty, R., Quille, K., Nolan, K., O'Farrell, A., & Becker, B. A. (2023). CS Outreach to Inform Secondary School Students' Perceptions of CS: Initial Findings. *Proceedings of the 18th WiPSCE Conference on Primary and Secondary Computing Education Research*, 1–6. <https://doi.org/10.1145/3605468.3609758>
- Ocak, C., Walters, K., & Kopcha, T. J. (2024). Fostering Equity in Computer Science Education: Principles from a Feminist Standpoint Conceptual Framework. *Tech Trends*. <https://doi.org/10.1007/s11528-024-00982-x>
- O'Hara, M., Lindner, R., Morley, R., Carney-Rees, H., & Compton, M. (2025). Rekindling the joy of learning. *Journal of Learning Development in Higher Education*, (35). <https://doi.org/10.47408/jldhe.vi35.1569>
- Okoko, J. M., Tunison, S., & Walker, K. D. (2023). *Varieties of qualitative research methods: Selected contextual perspectives*. Springer.
- Oleson, A., Xie, B., Salac, J., Everson, J., Kivuva, F. M., & Ko, A. J. (2022). A Decade of Demographics in Computing Education Research: A Critical Review of Trends in Collection, Reporting, and Use. *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1*, 1, 323–343. <https://doi.org/10.1145/3501385.3543967>
- Omidire, M. F., Aluko, F. R., & Mampane, M. R. (2021). Promoting the Joy of Teaching and Learning in a Diverse World. *South African Journal of Higher Education*, 35(5), 216–233. <https://doi.org/10.20853/35-5-3861>
- Ozer, E. J., Abraczinskas, M., Duarte, C., Mathur, R., Ballard, P. J., Gibbs, L., Olivas, E. T., Bewa, M. J., & Afifi, R. (2020). Youth Participatory Approaches and Health Equity: Conceptualization and Integrative Review. *American Journal of Community Psychology*, 66(3-4), 267–278. <https://doi.org/10.1002/ajcp.12451>
- Padiyath, A., & Nelson-Fromm, T. (2026). Reflecting on Thematic Analysis in Computer Science Education Research: A Field Guide for Researchers and Reviewers. *Proceedings of the 57th ACM Technical Symposium on Computer Science Education V.1*, 1, 790–796. Retrieved February 26, 2026, from <https://dl.acm.org/doi/10.1145/3770762.3772512>
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.
- Papert, S., & Harel, I. (1991). Situating Constructionism. *constructionism*, 36(2), 1–11.
- Paris, D. (2012). Culturally Sustaining Pedagogy: A Needed Change in Stance, Terminology, and Practice. *Educational Researcher*, 41(3), 93–97. <https://doi.org/10.3102/0013189X12441244>

- Parker, B., & Hankins, J. (2002). The Joys and Sorrows of Teaching Computer Literacy Online [ERIC Number: ED475945]. *National Educational Computing Conference Proceedings*. Retrieved October 17, 2025, from <https://eric.ed.gov/?id=ED475945>
- Parry, S. C. (1996). Feminist Pedagogy and Techniques for the Changing Classroom. *Women's Studies Quarterly*, 24(3/4), 45–54. Retrieved March 8, 2026, from <https://www.jstor.org/stable/40004358>
- Parvin, N., & Rouse, R. (2024). Feminist Philosophical Toys: Playful Companions and Live Theorization. *Hypatia*, 1–27. <https://doi.org/10.1017/hyp.2023.123>
- Parvin, N., Rouse, R., Alvarez, D., Haghani, S., Clark, S., Gaskins, N. R., Sullivan, A., Mergil, E., Pelizari, J., Anupam, A., Casula, P., & Gupta, S. (2022). Mess and Making Matters in Feminist Teaching. *Catalyst: Feminism, Theory, Technoscience*, 8(1). <https://doi.org/10.28968/cftt.v8i1.37713>
- Patitsas, E., Craig, M., & Easterbrook, S. (2014). A historical examination of the social factors affecting female participation in computing. *Proceedings of the 2014 conference on Innovation & technology in computer science education*, 111–116. <https://doi.org/10.1145/2591708.2591731>
- Patton, M. Q. (1999). Enhancing the quality and credibility of qualitative analysis. *Health Services Research*, 34(5 Pt 2), 1189–1208. Retrieved March 18, 2024, from <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1089059/>
- Patton, M. Q. (2015). *Qualitative research & evaluation methods: Integrating theory and practice* (Fourth edition.). SAGE Publications, Inc.
- Payne, W. C., Bergner, Y., West, M. E., Charp, C., Shapiro, R. B., Szafir, D. A., Taylor, E. V., & DesPortes, K. (2021). danceON: Culturally Responsive Creative Computing. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–16. <https://doi.org/10.1145/3411764.3445149>
- Pearson, T., & Leggett-Robinson, P. (2024). Unpacking the Unique Role of Black Women Computer Science Educators. *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, 1042–1048. <https://doi.org/10.1145/3626252.3630957>
- Peña, J., Hanrahan, B. V., Rosson, M. B., & Cole, C. (2021). After-Hours Learning: Workshops for Professional Women to Learn Web Development. *ACM Trans. Comput. Educ.*, 21(2), 15:1–15:31. <https://doi.org/10.1145/3446964>
- Peppler, K., Keune, A., Bender, S., & Yankova, N. (2025). Materialized Action: Reformulating the “Doing of” Math Through Fiber Crafting. *Cognition and Instruction*, 43(3), 175–200. <https://doi.org/10.1080/07370008.2025.2485070>
- Peppler, K., & Thompson, N. (2024). Tools and materials as non-neutral actors in STEAM education. *Journal of the Learning Sciences*, 0(0), 1–38. <https://doi.org/10.1080/10508406.2024.2380694>

- Perez, M. S., Lewis, C. M., & Hinze-Pifer, R. (2025). Analyzing High School CS Teachers' Likelihood of Attrition. *Proceedings of the 2025 ACM Conference on International Computing Education Research V.1*, 41–50. <https://doi.org/10.1145/3702652.3744215>
- Petelka, J., Finn, M., Roesner, F., & Shilton, K. (2022). Principles Matter: Integrating an Ethics Intervention into a Computer Security Course. *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1, 1*, 474–480. <https://doi.org/10.1145/3478431.3499275>
- Pinkard, N., Erete, S., Martin, C. K., & McKinney de Royston, 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>
- Porter, L., & Simon, B. (2013). Retaining nearly one-third more majors with a trio of instructional best practices in CS1. *Proceeding of the 44th ACM technical symposium on Computer science education*, 165–170. <https://doi.org/10.1145/2445196.2445248>
- Priniski, S. J., Hecht, C. A., & Harackiewicz, J. M. (2018). Making Learning Personally Meaningful: A New Framework for Relevance Research. *The Journal of Experimental Education*, 86(1), 11–29. <https://doi.org/10.1080/00220973.2017.1380589>
- Ramnath, S., & Hoover, J. H. (2016). Enhancing Engagement by Blending Rigor and Relevance. *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, 108–113. <https://doi.org/10.1145/2839509.2844554>
- Rankin, Y. A., Erete, S., Thomas, J. O., & Pinkard, N. (2024). The Choice is Yours: Intersectional Studies versus Studies of Intersectional Populations in Computing Education Research. *Proceedings of the 55th ACM Technical Symposium on Computer Science Education*, 1098–1104. <https://doi.org/10.1145/3626252.3630942>
- Rankin, Y. A., Thomas, J. O., & Erete, S. (2021a). Real Talk: Saturated Sites of Violence in CS Education. *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, 802–808. <https://doi.org/10.1145/3408877.3432432>
- Rankin, Y. A., Thomas, J. O., & Erete, S. (2021b). Black Women Speak: Examining Power, Privilege, and Identity in CS Education. *ACM Transactions on Computing Education*, 21(4), 1–31. <https://doi.org/10.1145/3451344>
- Ren, X. (2022). Adopting Feminist Pedagogy in Computer Science Education to Empower Underrepresented Populations: A Critical Review. *TechTrends*, 66(3), 459–467. <https://doi.org/10.1007/s11528-022-00728-7>

- Renninger, K. A., & Hidi, S. E. (2019). Interest Development and Learning. In K. A. Renninger & S. E. Hidi (Eds.), *The Cambridge Handbook of Motivation and Learning* (pp. 265–290). Cambridge University Press. <https://doi.org/10.1017/9781316823279.013>
- Resnick, M. (2020, June). Designing for Wide Walls. Retrieved January 10, 2024, from <https://mres.medium.com/designing-for-wide-walls-323bdb4e7277>
- Richard, G. T., & Kafai, Y. B. (2016). Blind Spots in Youth DIY Programming: Examining Diversity in Creators, Content, and Comments within the Scratch Online Community. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 1473–1485. <https://doi.org/10.1145/2858036.2858590>
- Riley, D. (2017). Rigor/Us: Building Boundaries and Disciplining Diversity with Standards of Merit. *Engineering Studies*, 9(3), 1–17. <https://doi.org/10.1080/19378629.2017.1408631>
- Riley, D. (2019). Developing a Liberative Pedagogy in Engineering. In N. Kellam, B. Coley, & A. Boklage (Eds.), *Transformative Teaching: A Collection of Stories of Engineering Faculty's Pedagogical Journeys* (pp. 9–25). Springer International Publishing. https://doi.org/10.1007/978-3-031-79406-3_2
- Roldan, W., Lee, K. J., Nguyen, K., Berhe, L., & Yip, J. (2022). Disrupting Computing Education: Teen-Led Participatory Design in Libraries. *ACM Transactions on Computing Education*, 22(3), 1–33. <https://doi.org/10.1145/3484494>
- Roque, R. (2023). Imagining Alternative Visions of Computing: Photo-Visuals of Material, Social, and Emotional Contexts from Family Creative Learning. *Proceedings of the 22nd Annual ACM Interaction Design and Children Conference*, 68–81. <https://doi.org/10.1145/3585088.3589235>
- Roque, R. (2024, June). Joy as an Aesthetic: Reflections on Designing Transformative Learning Experiences. Retrieved February 21, 2025, from <https://www.creativecommunities.group/blog/2024/6/5/joy-as-an-aesthetic-reflections-on-designing-transformative-learning-experiences>
- Roque, R., Tamashiro, M. A., McConnell, K., & Granados, J. (2021). Opportunities and Limitations of Construction Kits in Culturally Responsive Computing Contexts: Lessons from ScratchJr and Family Creative Learning. *Proceedings of the 20th Annual ACM Interaction Design and Children Conference*, 246–256. <https://doi.org/10.1145/3459990.3460728>
- Rosenblum, L. M., & Harris, L. (2019). Passing Through: Feminist Digital Pedagogy and Failure in the General Education Classroom. *Transformations: The Journal of Inclusive Scholarship and Pedagogy*, 29(1), 30–48. <https://doi.org/10.5325/trajincschped.29.1.0030>
- Ross, M. S. (2023). Let's have that Conversation: How Limited Epistemological Beliefs Exacerbates Inequities and will Continue to be a Barrier to Broadening Participation. *ACM Trans. Comput. Educ.*, 23(2), 17:1–17:4. <https://doi.org/10.1145/3578270>

- Rouvrais, S., & Kanellos, I. (2011). Facing computer science misconceptions: An introductory course based on historical strands and career paths at a glance. *2011 Frontiers in Education Conference (FIE)*, F4G–1–F4G–7. <https://doi.org/10.1109/FIE.2011.6142901>
- Ryoo, J., Goode, J., & Margolis, J. (2015). It takes a village: Supporting inquiry- and equity-oriented computer science pedagogy through a professional learning community. *Computer Science Education*, 25(4), 351–370. <https://doi.org/10.1080/08993408.2015.1130952>
- Ryoo, J. J. (2019a). “Laughter is the Best Medicine”: 177 Pedagogies of humor and joy that support critical thinking and communicative competence. In *Deeper Learning, Dialogic Learning, and Critical Thinking*. Routledge.
- Ryoo, J. J. (2019b). Pedagogy that Supports Computer Science for All. *ACM Trans. Comput. Educ.*, 19(4), 36:1–36:23. <https://doi.org/10.1145/3322210>
- Ryoo, J. J. (2025). Embracing Pedagogies of Joy With Joy: A Response to Scipio, Greenberg, Keifert, and Lee [eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/sce.70003>]. *Science Education*, n/a(n/a). <https://doi.org/10.1002/sce.70003>
- Ryoo, J. J., & Blunt, T. (2024). “Show Them the Playbook That These Companies Are Using”: Youth Voices about Why Computer Science Education Must Center Discussions of Power, Ethics, and Culturally Responsive Computing. *ACM Trans. Comput. Educ.*, 24(3), 34:1–34:21. <https://doi.org/10.1145/3660645>
- Ryoo, J. J. (2013). *Pedagogy Matters: Engaging Diverse Students as Community Researchers in Three Computer Science Classrooms* [Doctoral dissertation, UCLA]. Retrieved October 17, 2025, from <https://escholarship.org/uc/item/1qj8c044>
- Sánchez-Calle, A., & Velázquez-Iturbide, J. A. (1991). Fun, rigour and pragmatism in functional programming. *ACM SIGCSE Bulletin*, 23(3), 11–16. <https://doi.org/10.1145/126459.126464>
- Schneider, D. K., Boufflers, L., & Benetos, K. (2020). Computerized embroidery for identity building.
- Scipio, D., Greenberg, D., Keifert, D. T., & Lee, S. J. (2025). Pedagogies of Joy;) A Leap to Joy-Centered Critical Design. *Science Education*, 110, 313–332. <https://doi.org/https://doi.org/10.1002/sce.21983>
- Scott, K. A., Sheridan, K. M., & Clark, K. (2015). Culturally responsive computing: A theory revisited. *Learning, Media and Technology*, 40(4), 412–436. <https://doi.org/10.1080/17439884.2014.924966>
- Searle, K., & Kafai, Y. (2015). Boys’ Needlework: Understanding Gendered and Indigenous Perspectives on Computing and Crafting with Electronic Textiles. *Proceedings of the Eleventh Annual International Conference on International Computing Education Research*, 31–39. <https://doi.org/10.1145/2787622.2787724>
- Searle, K. A., & Kafai, Y. B. (2015). Culturally Responsive Making with American Indian Girls: Bridging the Identity Gap in Crafting and Computing with Electronic Textiles. *Proceedings of the Third Conference on GenderIT*, 9–16. <https://doi.org/10.1145/2807565.2807707>

- Seigfried, C. H. (2002). *Feminist interpretations of John Dewey*. Pennsylvania State University Press.
- Shank, M. J. (2000). Striving for Educational Rigor: Acceptance of Masculine Privilege. In *Masculinities at School* (pp. 213–230). SAGE Publications, Inc. <https://doi.org/10.4135/9781452225548.n9>
- Shaw, M. S. (2023). *Reimagining Adolescent Black Girls' Computing Futures Using Black Feminist-Womanist Restorying Through Design Methodologies* [Doctorate]. University of Pennsylvania. Retrieved February 27, 2025, from <https://www.proquest.com/docview/2854997414/abstract/517882CAB0304E00PQ/1>
- Shaw, M. S., Coleman, J. J., Thomas, E. E., & Kafai, Y. B. (2023). Restorying a Black girl's future: Using womanist storytelling methodologies to reimagine dominant narratives in computing education. *Journal of the Learning Sciences*, 32(1), 52–75. <https://doi.org/10.1080/10508406.2023.2179847>
- Shaw, M. S., Fields, D. A., & Kafai, Y. B. (2020). Leveraging local resources and contexts for inclusive computer science classrooms: Reflections from experienced high school teachers implementing electronic textiles. *Computer Science Education*, 30(3), 313–336. <https://doi.org/10.1080/08993408.2020.1805283>
- Sherif, E. (2025). Centering Students' Needs to Create Culturally Responsive Computing Curricula. *Proceedings of the 2025 ACM Conference on International Computing Education Research V.2*, 36–37. <https://doi.org/10.1145/3702653.3744303>
- Sherif, E., Everson, J., Kivuva, F. M., Kirdani-Ryan, M., & Ko, A. J. (2024). Exploring the Impact of Assessment Policies on Marginalized Students' Experiences in Post-Secondary Programming Courses. *Proceedings of the 2024 ACM Conference on International Computing Education Research - Volume 1*, 1, 233–245. <https://doi.org/10.1145/3632620.3671100>
- Shrewsbury, C. M. (1993). What Is Feminist Pedagogy? *Women's Studies Quarterly*, 21(3/4), 8–16. Retrieved March 8, 2026, from <https://www.jstor.org/stable/40022001>
- Slaton, A. E., & Riley, D. (2024). Rage-ography: Rigor, Anti-wokeness, and Technoviolence. *WSQ: Women's Studies Quarterly*, 52(1-2), 269–289. <https://doi.org/10.1353/wsqr.2024.a924321>
- Smith, G. (2024). Pairing Ungrading with Project-Based Learning in CS1 for Inherently Flexible Course Design. *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, 1265–1271. <https://doi.org/10.1145/3626252.3630903>
- Spieler, B., & Krnjic, V. (2021). Creative, Engaging, and Playful Making-Activities with Smartphones and Embroidery Machines. *FabLearn Europe / MakeEd 2021 - An International Conference on Computing, Design and Making in Education*, 1–4. <https://doi.org/10.1145/3466725.3466745>
- Spieler, B., & Schifferle, T. (2022). Maker-Education: Interdisciplinary Computer Science Activities. *Informatics in Schools*.
- Stahl, N. A., & King, J. R. (2020). Understanding and Using Trustworthiness in Qualitative Research. *Journal of Developmental Education*, 44(1), 26–28.

- Stommel, J., Friend, C., Morris, S. M., Rorabaugh, P., Rheingold, H., Watters, A., Davidson, C., Stewart, B., & Benjamin, R. (2020). *Critical Digital Pedagogy: A Collection*. Hybrid Pedagogy Inc.
- Sunday, A. O., Agbo, F. J., & Suhonen, J. (2024). Co-design Pedagogy for Computational Thinking Education in K-12: A Systematic Literature Review. *Technology, Knowledge and Learning*. <https://doi.org/10.1007/s10758-024-09765-y>
- Sunday, A. O., Agbo, F. J., Suhonen, J., Jormanainen, I., & Tukiainen, M. (2025). Co-designing to develop computational thinking skills in Nigeria K-12 using scratch. *Education and Information Technologies*. <https://doi.org/10.1007/s10639-025-13386-y>
- Tamboukou, M. (2018). The joy of learning: Feminist materialist pedagogies and the freedom of education. *Educational Philosophy and Theory*, 50(9), 868–877. <https://doi.org/10.1080/00131857.2017.1396213>
- Tissenbaum, M., Weintrop, D., Holbert, N., & Clegg, T. (2021). The case for alternative endpoints in computing education. *British Journal of Educational Technology*, 52(3), 1164–1177. <https://doi.org/10.1111/bjet.13072>
- Tofel-Grehl, C., Fields, D., Searle, K., Maahs-Fladung, C., Feldon, D., Gu, G., & Sun, C. (2017). Electrifying Engagement in Middle School Science Class: Improving Student Interest Through E-textiles. *Journal of Science Education and Technology*, 26(4), 406–417. <https://doi.org/10.1007/s10956-017-9688-y>
- Trim, M., & Gulley, P. (2023). Imagining, Generating, and Creating Communication as Feminist Pedagogical Method for Teaching Computing Ethics. *Proceedings of the 41st ACM International Conference on Design of Communication*, 206–209. <https://doi.org/10.1145/3615335.3623037>
- Turkle, S., & Papert, S. (1990). Epistemological Pluralism and the Revaluation of the Concrete. *Signs*, 16(1), 128–157. <https://www.jstor.org/stable/3174610>
- Vakil, S., & McKinney de Royston, M. (2022). Youth as philosophers of technology. *Mind, Culture, and Activity*, 29(4), 336–355. <https://doi.org/10.1080/10749039.2022.2066134>
- Vea, T., & Jaber, L. Z. (2025). Centering Affect and Emotion Toward Dignity and Justice in Science Education. *Science Education*, n/a(n/a). <https://doi.org/10.1002/sce.70028>
- Vella, K., Klarkowski, M., Turkay, S., & Johnson, D. (2020). Making friends in online games: Gender differences and designing for greater social connectedness. *Behaviour & Information Technology*, 39(8), 917–934. <https://doi.org/10.1080/0144929X.2019.1625442>
- Waller, A. (2005a). What is feminist pedagogy and how can it be used in CSET education? *Proceedings Frontiers in Education 35th Annual Conference*. <https://doi.org/10.1109/FIE.2005.1612155>
- Waller, A. (2005b). Work in progress - Feminist research methodologies: Why, what, and how. *Proceedings Frontiers in Education 35th Annual Conference*, F4H–20. <https://doi.org/10.1109/FIE.2005.1612158>

- Wang, J., Hejazi Moghadam, S., & Tiffany-Morales, J. (2017). Social Perceptions in Computer Science and Implications for Diverse Students. *Proceedings of the 2017 ACM Conference on International Computing Education Research*, 47–55. <https://doi.org/10.1145/3105726.3106175>
- Wang, J., Hong, H., Ravitz, J., & Hejazi Moghadam, S. (2016). Landscape of K-12 Computer Science Education in the U.S.: Perceptions, Access, and Barriers. *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, 645–650. <https://doi.org/10.1145/2839509.2844628>
- Weaver-Hightower, M. B. (2020). Critical Theories and Methods in Gender and Higher Education. In *The Wiley Handbook of Gender Equity in Higher Education* (pp. 469–492). John Wiley & Sons, Ltd. <https://doi.org/10.1002/9781119257639.ch24>
- Webb, L., Allen, M. W., & Walker, K. L. (2002). Feminist Pedagogy: Identifying Basic Principles. (the Scholarship of Teaching and Learning). *Academic exchange quarterly*. Retrieved March 9, 2026, from <https://www.semanticscholar.org/paper/Feminist-Pedagogy%3A-Identifying-Basic-Principles.-of-Webb-Allen/68fb914d06d90276b99102789a1a0f70d17d415f>
- Weiler, K. (1991). Freire and a Feminist Pedagogy of Difference. *Harvard Educational Review*, 61(4), 449–475. <https://doi.org/10.17763/haer.61.4.a102265jl68rju84>
- Wigginton, B., & Lafrance, M. N. (2019). Learning critical feminist research: A brief introduction to feminist epistemologies and methodologies. *Feminism & Psychology*. <https://doi.org/10.1177/0959353519866058>
- Wilkens, L. (2003). The joy of teaching with C#. *J. Comput. Sci. Coll.*, 19(2), 254–264.
- Woglom, J. F., & Jones, S. (2024). The Jam: Speculative-Mutant Pedagogies, Aesthetic Education Theory, and Becoming Joy with Children in a What If World. *Teachers College Record*, 126(11-12), 171–192. <https://doi.org/10.1177/01614681251317579>
- Wohlwend, K., & Peppler, K. (2015). All rigor and no play is no way to improve learning. *Phi Delta Kappan*, 96(8), 22–26. <https://doi.org/10.1177/0031721715583957>
- Wolz, U., Auschauer, M., & Mayr-Stalder, A. (2019a). Code crafting with turtl stitch. *ACM SIGGRAPH 2019 Studio*, 1–2. <https://doi.org/10.1145/3306306.3328009>
- Wolz, U., Auschauer, M., & Mayr-Stalder, A. (2019b). Programming embroidery with turtl stitch. *ACM SIGGRAPH 2019 Studio*, 1–2. <https://doi.org/10.1145/3306306.3328002>
- Wolz, U., Charles, G., Feire, L., & Nicolson, E. (2018). Code Crafters Curriculum: A Textile Crafts Approach To Computer Science (Abstract Only). *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 1055. <https://doi.org/10.1145/3159450.3162360>
- Woodman, M., Davies, G., & Holland, S. (1996). The joy of software—starting with objects. *Proceedings of the twenty-seventh SIGCSE technical symposium on Computer science education*, 88–92. <https://doi.org/10.1145/236452.236515>

- Worsley, M., & Bar-El, D. (2022). Inclusive Making: Designing tools and experiences to promote accessibility and redefine making. *Computer Science Education*, 32(2), 155–187. <https://doi.org/10.1080/08993408.2020.1863705>
- Wyse, S. A., & Soneral, P. A. G. (2018). “Is This Class Hard?” Defining and Analyzing Academic Rigor from a Learner’s Perspective. *CBE Life Sciences Education*, 17(4), ar59. <https://doi.org/10.1187/cbe.17-12-0278>
- Yadav, A., & Berges, M. (2019). Computer Science Pedagogical Content Knowledge: Characterizing Teacher Performance. *ACM Trans. Comput. Educ.*, 19(3), 29:1–29:24. <https://doi.org/10.1145/3303770>
- Yadav, A., Heath, M., & Hu, A. D. (2022). Toward justice in computer science through community, criticality, and citizenship. *Communications of the ACM*, 65(5), 42–44. <https://doi.org/10.1145/3527203>
- Yardi, S., & Bruckman, A. (2007). What is computing? bridging the gap between teenagers’ perceptions and graduate students’ experiences. *Proceedings of the third international workshop on Computing education research*, 39–50. <https://doi.org/10.1145/1288580.1288586>
- Yates, J., & Plagnol, A. C. (2022). Female computer science students: A qualitative exploration of women’s experiences studying computer science at university in the UK. *Education and Information Technologies*, 27(3), 3079–3105. <https://doi.org/10.1007/s10639-021-10743-5>
- Zhou, N., Nguyen, H., Fischer, C., Richardson, D., & Warschauer, M. (2020). High School Teachers’ Self-efficacy in Teaching Computer Science. *ACM Transactions on Computing Education*, 20(3), 23:1–23:18. <https://doi.org/10.1145/3410631>