

Student Experiences of Joyful Secondary CS Classrooms

Jayne Everson
everjay@uw.edu
University of Washington
Seattle WA USA

Rotem Landesman
roteml@uw.edu
University of Washington
Seattle WA USA

Janet Jiang
jcjiang@uw.edu
University of Washington
Seattle WA USA

F. Megumi Kivuva
megumik@uw.edu
University of Washington
Seattle WA USA

Eman Sherif
emans@uw.edu
University of Washington
Seattle WA USA

Amy J. Ko
ajko@uw.edu
University of Washington
Seattle WA USA

Abstract

Background: Prior research argues that positive learning environments support student learning, belonging, and the desire to continue learning. In order to create joyful learning environments, we must first study what characterizes such learning spaces. However, prior research has not yet asked students what makes CS learning environments joyful.

Objective: In this study, we sought to understand students' joyful experiences in secondary CS classrooms and the elements of the learning environments that fostered those experiences.

Method: We interviewed 8 students who had joyful CS experiences about those experiences, the environments in which they occurred, and about the pedagogies their teachers used. We then interviewed 5 of their teachers to better understand the environments and pedagogies that cultivated joyful CS experiences.

Findings: We found 4 elements in common in each student's experience: teachers worked to *relate to students*, *prioritize a relationship with learning*, *frame CS with purpose*, and *reify a joyful CS experience*.

Implications: These findings offer the opportunity for researchers and educators to reconsider pedagogies and design learning environments that engender joyful CS experiences for students. This will better enable students to build relationships with CS.

CCS Concepts

• **Social and professional topics** → **Computing education**.

Keywords

Joy, Education, Secondary, Pedagogy

ACM Reference Format:

Jayne Everson, Rotem Landesman, Janet Jiang, F. Megumi Kivuva, Eman Sherif, and Amy J. Ko. 2026. Student Experiences of Joyful Secondary CS Classrooms. In *Proceedings of the ACM Conference on International Computing Education Research Vol.1 (ICER 2026 Vol. 1)*, August 11–14, 2026, Uppsala, Sweden. ACM, New York, NY, USA, 14 pages. <https://doi.org/10.1145/3765964.3811650>



This work is licensed under a Creative Commons Attribution 4.0 International License. *ICER 2026 Vol. 1, Uppsala, Sweden*

© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2203-5/2026/08
<https://doi.org/10.1145/3765964.3811650>

1 Introduction

All classrooms, including Computer Science (CS) classrooms, should be places that value humanity and center the learners' experience [16]. These environments should foster a sense of belonging [9], as prior research makes clear that positive experiences encourage learners to keep learning [23, 24]. One word to describe belonging and the desire to continue learning is *joy* [16].

Scholars are already engaged in understanding joy in learning, giving talks about joy [38], and designing pedagogy for joy [18]. There has been some research studying joy in CS in particular. Sometimes instructors seek to design for their own joy [46]. Others learned that not all formats of learning computing are joyful for all students [33]. Some instructors designed curriculum intending to produce joy and found that it encouraged broader participation [1, 11, 12, 14, 25, 30]. Some have designed for joy in computing for particular populations; for example, centering connection in elementary CS learning [20].

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. 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 have much to teach us about what pedagogies and values led to their joyful experiences. After we spoke to the students, we asked them to connect us with their educators to add an additional lens to this study.

2 Related Work

There are many definitions for joy. For example, Roque frames joy as something dynamic that can coexist with struggle and grief [38]. It 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” [38]. Alternatively, Gholdie Muhammad focuses particularly on Black joy and defines it as “enkindl[ing] the most sublime emotions that can animate the human soul” [29]. We discuss how these broader conceptions of joy have shown up in research on learning.

2.1 Joyful Pedagogy

There is compelling prior work that suggests that joy should be a meaningful part of learning, and therefore a pedagogical goal. Approaching this work from the feminist materialist 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 [45]. 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, creation, and limited adult intervention were essential in creating this joyful cacophonous environment to learn to make music [47].

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” [44]. Another joyful pedagogy text argues for three factors: agency, community, and creativity, but again centers the teacher’s joy [4]. These works suggest that it is sometimes essential that teachers and instructors experience joy in teaching, but they do not connect this to students’ experiences of joyful learning.

Other prior work in higher education has broadly examined the phenomenon 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 [32]. 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 [13]. 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 [31].

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 [29]. 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 [18]. 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 [42]. Responding to the POY, Ryoo calls for more work that focuses on the voices and perspectives of young people [39]. 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.” [41]. This work found that humor also supported critical thinking and communicative competence.

2.2 Joy in CS education

In CS education research, we see a variety of definitions of joy, 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. For example, the *Beauty and Joy of Computing* [11, 12], which is “designed to appeal to a wide range of students, including women and underrepresented minorities,” was piloted at Berkeley and adopted in a variety of ways for AP courses [14], underrepresented students [1], and as a new AP course [25]. It has also been used to design new programming languages [30], and to foster engagement in a first year engineering course [28]. Sometimes instructors center their own joy in teaching. This might be to advocate for a particular programming language which they find clear and useful to teach [46]. While in other prior work, instructors designed online platforms for their own joy [33]. Furthermore, instructors designed for what they perceived students should find joyful in online learning [48]. In another example, Dijkstra (of the famous algorithm) gave a talk extolling the importance of formal methods in computing where he finds that, “[t]eaching to unsuspecting youngsters the effective use of formal methods is one of the joys of life because it is so extremely rewarding” [7].

Other times, practitioners design for joy in particular populations, like elementary students [20]. Here, instructors designed activities for elementary students to experience community and the joy of computing. However, this work measures joy from instructors’ perception, rather than students’ perceptions of their own joy. Practitioners also design in order to include marginalized populations needs. Lee et al. designed integrated math and science curriculum for Latinx students [21], and found significant improvements in both their knowledge and their perceptions of computer science. Prior work centering Black joy in computing environments by Ivey et al. advocate for abolitionist teaching practices in secondary CS courses by centering the voices of Black educators to center Black joy [17].

Some research examines how important and difficult it is to create spaces that center joy. Additionally, Jones and Melo call for more critical conversations in CS education by imagining alternative visions of CS that center student joy and liberation [19]. In a study examining the practices of creative coding instructors, 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 [27].

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 [26]. Additionally, Porter et al. included pair programming, media computation, and peer instruction and found increases in retention and enjoyment [36]. Chavez shared that limiting course creep and examining deadlines improved outcomes in a data structures course [5].

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 CS from the student perspective.

3 Methods

We interviewed participants who self-identified as having joyful secondary computing experiences [3] to better understand those experiences. We also asked student participants to identify teachers in their joyful experiences, and interviewed those teachers. We conducted qualitative research using Braun and Clarke’s 6 steps for Thematic Analysis [3, 34]. This study was IRB reviewed and exempt at the researchers’ home institution.

3.1 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 the interviews. Because joy is an emotional experience, joy can only be identified by the individual experiencing it. We did not approach this work with a definition of a “joyful experience,” we left it to individuals sharing their interpretations of joy rather than defining it for them.

3.2 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 included

students who had joyful secondary computer science environments. Our exclusion criteria were 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 1.

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

3.4 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

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)	Advanced Placement CS, Avanced Topics TEALS classroom (Teacher A was an industry support instructor)
Student 3	Teacher A	Advanced Placement CS, Avanced Topics TEALS classroom (Teacher A was an industry support educator)
Student 4	(teacher never replied)	Advanced Placement CS Principles
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 this student’s interest in CS into other coursework (Teacher D taught Math)
Student 7	Teacher C	Advanced Placement CS
Student 8	Teacher E	Advanced Placment CS A 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)	Advanced Placement CS Principles and Advanced Placment CS A

Table 1: Table of the students, joyful educators, and brief context of the environment of joyful CS

journaling about their own joyful computing environments and experiences of joy in computing. In each step of the analysis process, the team started with students’ joyful computing experiences, and then corroborated with teachers’ interviews. Using student transcripts, the team individually identified themes of environmental factors that were part of the student’s joyful 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 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.

4 Results

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 1 shows these 4 elements, aligned with growth in trust and student agency.

4.1 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.

4.1.1 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

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

with students was one of the hardest things for him to do, but 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

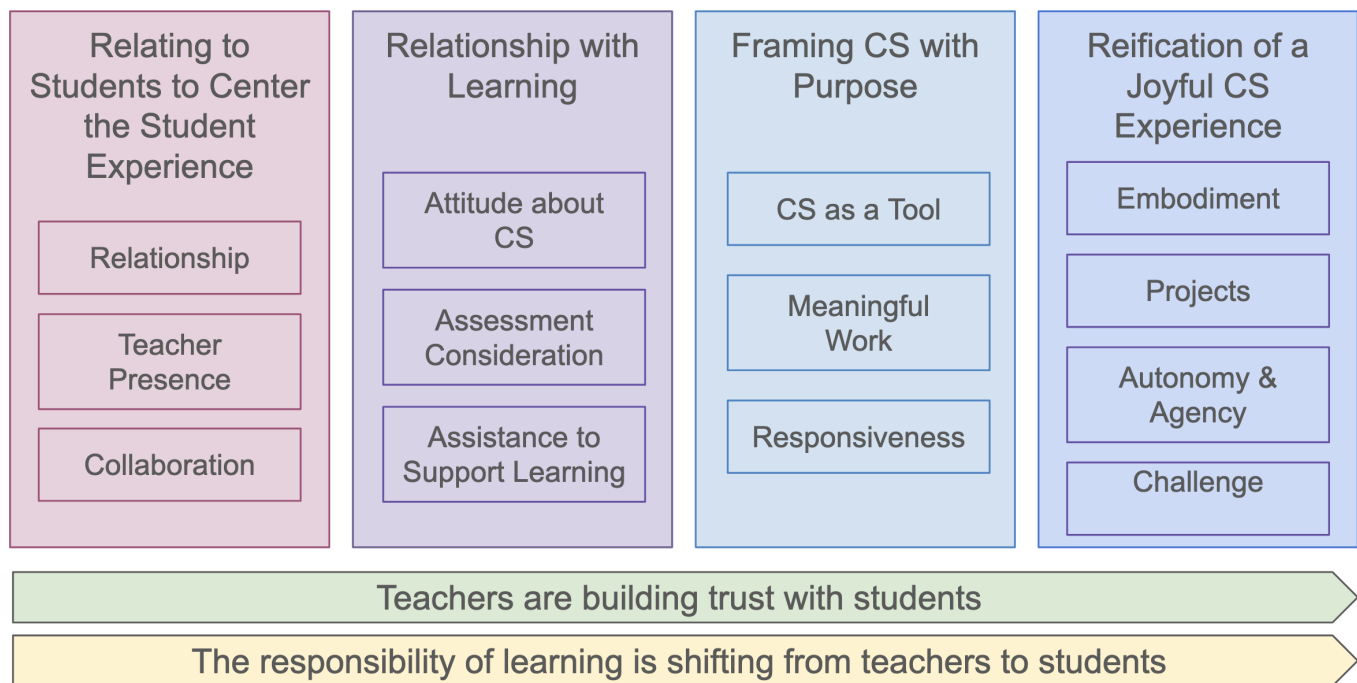


Figure 1: The research team conceptualized the reification of a joyful classroom with the above diagram of four elements of what Teachers were doing to create these joyful environments. Each of the four elements has several different components. Not all students mentioned each of the components, but they did mention more than one component from each of the four elements.

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!’”

4.1.2 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 make sure every student had the support they needed. Student 6 shared how their teachers knew they liked CS, and because they 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.

4.1.3 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 was not afraid to admit she did not know an 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.

4.2 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.

4.2.1 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 memory 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.

4.2.2 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 the 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?’ 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.

4.2.3 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 [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.

4.3 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.

4.3.1 *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.

4.3.2 *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 liked 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 having a joyful experience while learning, students had a deep understanding of complex material (which will be discussed in Section 4.4.4).

4.3.3 *Responsiveness.* Teachers were responsive and worked to incorporate students’ 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 4.4.3.

4.4 Reification of a Joyful CS Experience: Trust Established

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

4.4.1 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.

4.4.2 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 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.’”

4.4.3 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-based 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 B 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.*”

4.4.4 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 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 Discussion

We identified four elements of joyful computing environments. In each step of this research we started with the students’ joyful experiences and then corroborated those experiences with teachers’ perspectives. 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.

Each of the elements students’ identified as important part of their joyful experiences could be considered part of “good teaching.” However, this study sought to understand students’ joyful

computing experiences. Students were recruited because they had joyful experiences, and they shared those with the research team. It is not surprising that many of the environments that allowed for students’ joyful experiences to flourish also had good teachers, excellent pedagogy. In contrast, prior work often focuses on teacher and pedagogical aspects of joyful experiences, this work focuses on what students’ joyful experiences are.

The elements of joyful experiences we discussed in the results 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.

5.1 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. Joy was a collaborative effort that existed as a result of relationships in the classroom. Both teachers and students contributed to its existence.

Relationships between teachers, students, and peers. Prior work has conceptualized joy in rich ways [38]. 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 [40]. 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) [6]. The fact that these elements surfaced strengthens CRSP’s claims, and emphasizes 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 students’ needs around assessments [43], that students need flexibility and care from instructors in order to feel seen in the learning process.

5.2 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 [35] 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 were often motivated to learn complex content to achieve their goals. Students mentioned over and over again how CS allowed them to 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 [8]. 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 [37], identity development [22], and self-efficacy [2]. 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.

5.3 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 [16, 45], value student agency, and autonomy [10, 15, 47]. 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.

5.4 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. All of the interviews were retrospective, and findings depend on participants memory and interpretations of their experiences. 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 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 students’ 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. Further work is needed to understand what parts a teacher contributes to joy and what part a student brings to a joyful experience, and if like relationships, the burden of joy shifts from the teacher to the student. There is also further work needed to understand if a student has previous experiences or other background factors that contribute to their ability to have a joyful CS experience. Additionally, there is the opportunity once researchers have identified joy in a field that is so often perceived as dry and boring [8], to compare this joy

against pedagogy in other disciplines, and perhaps make schools more joyful overall.

Acknowledgments

Thank you to the teachers who are working to make K-12 CS a joyful experience for their students. This material is based upon work supported by the National Science Foundation under Grant No. 2417014, 2031265, 2100296, 2122950, 2023338658, 2024372363 and unrestricted gifts from Microsoft, Adobe, and Google.

References

- [1] Tiffany M Barnes. 2013. Type I: Collaborative Research: FRABJIOUS CS - Framing a Rigorous Approach to Beauty and Joy for Outreach to Underrepresented Students in Computing at Scale. 13 (2013), 46922. <https://ui.adsabs.harvard.edu/abs/2013nsf....1346922B> ADS Bibcode: 2013nsf....1346922B.
- [2] Rakhi Batra and Zahra Atiq. 2025. Emotions and Self-Efficacy of Undergraduate Computing and Engineering Students: A Systematic Literature Review. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1* (Pittsburgh PA USA, 2025-02-12). ACM, 88–94. doi:10.1145/3641554.3701904
- [3] Virginia Braun and Victoria Clarke. 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. 21, 1 (2021), 37–47. doi:10.1002/capr.12360
- [4] Eileen Kogl Camfield. 2025. *Joy-Centered Pedagogy in Higher Education: Uplifting Teaching and Learning for All*. Taylor & Francis. Google-Books-ID: 7RVEE-QAAQBAJ.
- [5] Victoria C. Chávez. 2023. How My Students and I (Re)Discovered the Joy of Computing in CS2. 14, 2 (2023), 36–39. doi:10.1145/3596918
- [6] Kalisha Davis, Shana V. White, Dinah Becton-Consuegra, and Allison Scott. 2021. Culturally Responsive-Sustaining Computer Science Education: A Framework. <https://www.kaporcenter.org/culturally-responsive-sustaining-computer-science-education-a-framework/>
- [7] Edsger W Dijkstra. [n. d.]. On the cruelty of really teaching computing science. ([n. d.]).
- [8] Jayne Everson, Rotem Landesman, F. Megumi Kivuva, and Amy J. Ko. 2025. Dreaming of Difference: Imagining the Future of CS Education Pedagogy with Secondary Students. In *Proceedings of the 2025 ACM Conference on International Computing Education Research V.1* (ICER '25). Association for Computing Machinery, New York, NY, USA, 394–406. doi:10.1145/3702652.3744203
- [9] Carol L. Fletcher and Jayce R. Warner. 2021. CAPE: a framework for assessing equity throughout the computer science education ecosystem. 64, 2 (2021), 23–25. doi:10.1145/3442373
- [10] Paulo Freire and Myra Bergman Ramos. 2014. *Pedagogy of the oppressed* (30th anniversary edition. ed.). Bloomsbury.
- [11] Daniel D. Garcia, Robb Cutler, Zachary Dodds, Eric Roberts, and Alison Young. 2009. Rediscovering the passion, beauty, joy, and awe: making computing fun again, continued. In *Proceedings of the 40th ACM technical symposium on Computer science education* (New York, NY, USA, 2009-03-04) (SIGCSE '09). Association for Computing Machinery, 65–66. doi:10.1145/1508865.1508889
- [12] Daniel D Garcia and Brian Harvey. 2011. Collaborative Research: Type 1: FRABJIOUS CS - Framing a Rigorous Approach to Beauty and Joy for Outreach to Underrepresented Students in Computing at Scale. 11 (2011), 38596. <https://ui.adsabs.harvard.edu/abs/2011nsf....1138596G> ADS Bibcode: 2011nsf....1138596G.
- [13] Alexander Hall and Brent Steele. 2024. Joy as subversive defiance. 20, 3 (2024), 232–246. doi:10.1177/17550882241284043
- [14] Brian Harvey. 2012. The Beauty and Joy of Computing: Computer Science for Everyone. (2012).
- [15] bell hooks. 2015. *Feminism is for everybody: passionate politics* ([second edition]. ed.). Routledge. doi:10.4324/9781315743189
- [16] bell hooks. 1994. *Teaching to transgress: education as the practice of freedom*. Routledge.
- [17] Allison Ivey, Stephany RunningHawk Johnson, Max Skorodinsky, Jimmy Snyder, and Joanna Goode. 2021. Abolitionist Computer Science Teaching: Moving from Access to Justice. In *2021 Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT)* (2021-05). 1–4. doi:10.1109/RESPECT51740.2021.9620652
- [18] Igor Jasinski and Tyson E. Lewis. 2016. Another Kind of Love in Education: Whatever Love. 47, 4 (2016), 429–441. doi:10.1007/s10780-016-9282-x
- [19] Stephanie T. Jones and Natalie Melo. 2020. 'Anti-blackness is no glitch': the need for critical conversations within computer science education. 27, 2 (2020), 42–46. doi:10.1145/3433134
- [20] Mike Karlin, Cristina Stephany, and Marcia Reed. 2023. Coding to Connect: Centering Joy and Community in Elementary Computer Science Education. 1, 1 (2023), 71–84. doi:10.46787/dialogue.v1i1.3327
- [21] Hakeoung Hannah Lee, Sylvia Celedón-Pattichis, Marios S. Pattichis, Amy Rae Johnson, Esteban Cantú, Irán Tovar, and Carlos A. LópezLeiva. 2023. Knowing and Enjoying: Expanding Latinx Students' Experiences With an Integrated Computer Science and Mathematics Curriculum. (2023). doi:10.3102/2012330
- [22] Colleen M. Lewis, Ruth E. Anderson, and Ken Yasuhara. 2016. "I Don't Code All Day": Fitting in Computer Science When the Stereotypes Don't Fit. In *Proceedings of the 2016 ACM Conference on International Computing Education Research* (Melbourne VIC Australia, 2016-08-25). ACM, 23–32. doi:10.1145/2960310.2960332
- [23] Jane Margolis, Rachel Estrella, Joanna Goode, Jennifer Jellison Holme, and Kimberly Nao. 2008. *Stuck in the Shallow End: Education, Race, and Computing*. The MIT Press. <http://www.jstor.org/stable/j.ctt1mtz830>
- [24] Jane Margolis and Allan Fisher. 2002. *Unlocking the Clubhouse: Women in Computing*. MIT Press. Google-Books-ID: StwGQw45YoEC.
- [25] June Mark and Kelsey Klein. 2019. Beauty and Joy of Computing: 2016-17 Findings from an AP CS Principles course. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (New York, NY, USA, 2019-02-22) (SIGCSE '19). Association for Computing Machinery, 627–633. doi:10.1145/3287324.3287375
- [26] Charlie McDowell, Linda Werner, Heather E. Bullock, and Julian Fernald. 2006. Pair programming improves student retention, confidence, and program quality. 49, 8 (2006), 90–95. doi:10.1145/1145287.1145293
- [27] Andrew M McNutt, Sam Cohen, and Ravi Chugh. 2025. Slowness, Politics, and Joy: Values That Guide Technology Choices in Creative Coding Classrooms. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2025-04-25) (CHI '25). Association for Computing Machinery, 1–16. doi:10.1145/3706598.3713472
- [28] Federico Meza, Nicolás Torres, and Andrea Vásquez. 2020. Fostering Engagement and Creativity through Programming: The Beauty and Joy of Computing in a First-year Engineering class. In *2020 39th International Conference of the Chilean Computer Science Society (SCCC)* (2020-11). 1–5. doi:10.1109/SCCC51225.2020.9281159
- [29] Gholnecsar E. Muhammad. 2022. Cultivating Genius and Joy in Education through Historically Responsive Literacy. 99, 3 (2022), 195–204. doi:10.58680/la202231623
- [30] Jens Mönig. 2020. Building Blocks for Powerful Ideas: Designing a Programming Language to Teach the Beauty and Joy of Computing. 45–46. doi:10.26439/ciis2019.5499
- [31] Maria O'Hara, Rebecca Lindner, Rachel Morley, Haf Carney-Rees, and Martin Compton. 2025. Rekindling the joy of learning. 35 (2025). doi:10.47408/jldhe.vi35.1569
- [32] M. F. Omidire, F. R. Aluko, and M. R. Mampane. 2021. Promoting the Joy of Teaching and Learning in a Diverse World. 35, 5 (2021), 216–233. doi:10.20853/35-5-3861
- [33] Brenda Parker and Judith Hankins. 2002. The Joys and Sorrows of Teaching Computer Literacy Online. In *National Educational Computing Conference Proceedings* (2002-06). For full text: <http://confreg.https://eric.ed.gov/?id=ED475945> ERIC Number: ED475945.
- [34] Michael Quinn Patton. 2015. *Qualitative research & evaluation methods: integrative theory and practice* (fourth edition. ed.). SAGE Publications, Inc.
- [35] Kylie Pappeler and Naomi Thompson. [n. d.]. Tools and materials as non-neutral actors in STEAM education. 0, 0 ([n. d.]), 1–38. doi:10.1080/10508406.2024.2380694 _eprint: <https://doi.org/10.1080/10508406.2024.2380694>.
- [36] Leo Porter and Beth Simon. 2013. Retaining nearly one-third more majors with a trio of instructional best practices in CS1. In *Proceeding of the 44th ACM technical symposium on Computer science education* (New York, NY, USA, 2013-03-06) (SIGCSE '13). Association for Computing Machinery, 165–170. doi:10.1145/2445196.2445248
- [37] K. Ann Renninger and Suzanne E. Hidi. 2019. Interest Development and Learning. In *The Cambridge Handbook of Motivation and Learning*, K. Ann Renninger and Suzanne E. Hidi (Eds.). Cambridge University Press, 265–290. doi:10.1017/9781316823279.013
- [38] Ricarose Roque. 2024. Joy as an Aesthetic: Reflections on Designing Transformative Learning Experiences. (2024). <https://www.creativecommunities.group/blog/2024/6/5/joy-as-an-aesthetic-reflections-on-designing-transformative-learning-experiences> AERA.
- [39] Jean J. Ryoo. 2025. Embracing Pedagogies of Joy With Joy: A Response to Scipio, Greenberg, Keifert, and Lee. (2025). doi:10.1002/sce.70003 _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/sce.70003>.
- [40] Jean Jinsun Ryoo. 2013. Pedagogy Matters: Engaging Diverse Students as Community Researchers in Three Computer Science Classrooms. <https://escholarship.org/uc/item/1qj8c044>
- [41] Jean J. Ryoo. 2019. "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. Num Pages: 17.
- [42] Déana Scipio, Day Greenberg, D. Teo Keifert, and Sarah J. Lee. 2025. Pedagogies of Joy: A Leap to Joy-Centered Critical Design. (2025). doi:10.1002/sce.21983 _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/sce.21983>.
- [43] Eman Sherif, Jayne Everson, F. Megumi Kivuva, Mara Kirdani-Ryan, and Amy J. Ko. 2024. Exploring the Impact of Assessment Policies on Marginalized Students'

- Experiences in Post-Secondary Programming Courses. In *Proceedings of the 2024 ACM Conference on International Computing Education Research - Volume 1* (New York, NY, USA, 2024-08-12) (*ICER '24, Vol. 1*). Association for Computing Machinery, 233–245. doi:10.1145/3632620.3671100
- [44] Kahlil Simpson. 2023. Imagining Joy. *Promising Pedagogies for Teacher Inquiry and Practice: Teaching Out Loud* (2023), 97.
- [45] Maria Tamboukou. 2018. The joy of learning: Feminist materialist pedagogies and the freedom of education. *50, 9* (2018), 868–877. doi:10.1080/00131857.2017.1396213 _eprint: <https://doi.org/10.1080/00131857.2017.1396213>.
- [46] Linda Wilkens. 2003. The joy of teaching with C#. *19, 2* (2003), 254–264.
- [47] James F. Woglom and Stephanie Jones. 2024. The Jam: Speculative-Mutant Pedagogies, Aesthetic Education Theory, and Becoming Joy with Children in a What If World. *126, 11* (2024), 171–192. doi:10.1177/01614681251317579
- [48] Mark Woodman, Gordon Davies, and Simon Holland. 1996. The joy of software—starting with objects. In *Proceedings of the twenty-seventh SIGCSE technical symposium on Computer science education* (New York, NY, USA, 1996-03-01) (*SIGCSE '96*). Association for Computing Machinery, 88–92. doi:10.1145/236452.236515

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009