# Constructing Computational Identities

## Exploring Constructionism and Identity in an Introductory High School Computer Science Course

Jacob Wolf
Harvard University
jwolf@g.harvard.edu

Jenny Han
Stanford University
jennyhan@cs.stanford.edu

Chris Proctor
University at Buffalo, SUNY
chrisp@buffalo.edu

Emma Brown
Independent Schools Foundation
eqbrown@isf.edu.hk

Paulo Blikstein
Teachers College
paulob@tc.columbia.edu

## ABSTRACT

Justifying universal K12 computer science (CS) education requires a clear articulation of why CS is valuable for all students. Drawing from Constructionist perspectives on the relationship between computers and cognition, we analyze interviews with secondary school students at the end of their time in a 2-year, constructionist CS course to better understand how these students' CS education affected their experiences with computing in their everyday lives. In doing so, we identify three types of experience where students begin to "re-see" technology: respect, curiosity, and an impetus to do more. We analyze these experiences using Constructionist perspectives on computer cultures, material intelligence, and liberatory pedagogy. Finally, we discuss how computational identity development is a unique process mediated by learners' participation in communities of practice and explore implications of this analysis for the goals of K12 CS education.

## CCS CONCEPTS

• **Social and professional topics**; • **Computing education**; • **K-12 education**;

## KEYWORDS

CS Education, Computational Identity, Communities of Practice

**ACM Reference Format:**
Jacob Wolf, Jenny Han, Chris Proctor, Emma Brown, and Paulo Blikstein. 2023. Constructing Computational Identities: Exploring Constructionism and Identity in an Introductory High School Computer Science Course. In *FabLearn / Constructionism 2023: Full and Short Research Papers (FLC 2023), October 07–11, 2023, New York City, NY, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3615430.3615446

## 1 1 INTRODUCTION

When Papert published Mindstorms in 1980, the world was dramatically shifting both socially and technologically. Constructionism sought to think about these shifts holistically: technological change was connected to cognitive change which was connected to social change. This alignment involved an ambitious, political goal of imagining how computers could help liberate our thinking from the "many cultural barriers [that] impede children from making scientific knowledge their own" [15, p. 4]. To accomplish this goal, Papert studied children's interactions with the computer cultures of the time to understand how they affected their thinking about mathematics and their relationship to the field.

In the context of the pandemic and recent innovations in artificial intelligence, we are arguably at another social and technological inflection point. Like Papert's moment in 1980, this moment challenges us to question what computing will mean for how people think. However, unlike Papert's moment, the ubiquity and increased power of personal computing has brought about the widespread computer cultures Papert only imagined. While Papert envisioned the opportunities for learning and for liberation this ubiquity would afford, the rise of disinformation, surveillance capitalism, and violent populism have demonstrated accompanying oppressive impacts of computers' effect on how people think.

Nevertheless, our moment is ripe for a reimagining of the relationship between computing and society: public scrutiny of technology companies is increasing, a new wave of computational participation is rising in creator-culture-focused digital spaces, and K12 schools across the world are becoming increasingly interested in and equipped for offering computer science (CS) education. As one step towards this re-imagining, Wing's [25] computational thinking has been proposed as a framework for understanding and solving computational problems. While this framework has been developed with cognitive, situated, and critical framings, we have yet to fully justify universal K12 CS education [13]. This justification must move beyond traditional notions of CS education exclusively leading to software engineering jobs and instead must articulate what CS can contribute to *all* learners' lives [20]. Making this justification requires us to investigate CS as a discipline while simultaneously investigating computing as an everyday experience in learners' lives [17]. This investigation will allow us to cultivate richer CS education which allows students to identify and challenge the dominant narratives within CS and within computing cultures more broadly. This work has already begun in CS education with projects like Shaw et al.'s [18] use of "restorying" to allow youth to situate their experiences with computing within broader cultural narratives and then challenge those narratives with new stories about who they are and what they can do.


*FLC 2023, October 07–11, 2023, New York City, NY, USA*

ACM ISBN 979-8-4007-0896-1/23/10…$15.00
https://doi.org/10.1145/3615430.3615446

Constructionism has contributed significantly to this work through its concern for learners' relationships with technology and the computer cultures in which they participate. Like Papert's "Mathland," we can study the effects of children stepping into the land of computing as they explore computational ideas. Though computing is nearly ubiquitous, we draw parallels with Papert's concerns that, while mathematical ideas exist throughout our worlds, they are often obscured. Similarly, most modern interfaces children use to access computer cultures are highly designed to reduce the visibility of the computational structures that underlie them. Silver [19] proposes the notion of a "constructive lens" that unveils these obscured structures. As learners develop this constructive lens, Silver documents moments of "re-seeing" the world which reveal previously unrecognized layers of abstraction and offer new inroads for manipulating components within those newly recognized layers.

In a previous paper, we explored how students in an introductory, Constructionist secondary school CS class began to "re-see" the computational artifacts in their everyday lives and how these perceptions influenced students' self-described relationships to computation, a construct we defined as "computational identity" [26]. With this paper, we extend this analysis by categorizing these re-seeing experiences into three types: respect for people who create computational artifacts, curiosity about the workings of those artifacts, and an impetus to do more to put their experiences with computation into practice. We then situate this analysis within a framework of Constructionism to suggest implications for CS education more broadly, specifically shaping how a universal K12 CS education could more successfully prepare learners to participate in the computational worlds that surround them.

# 2 BACKGROUND

This research project was developed around a Constructionist framework made up of three themes: computer cultures, material intelligence, and liberatory pedagogy [16]. These themes were used to design the Constructionist CS course at the center of this project and also contributed to our understanding of computational identity as researchers, curriculum designers, and educators. Below, we highlight the connections between our Constructionist framework and our definition of computational identity.

## 2.1 Computer cultures

Our definition of computational identity draws on past work exploring how students learn CS through their participation in computer cultures and how this participation influences the ways students think of themselves relative to CS [11]. Computer cultures were essential to Papert's definition of computing and participation in computer cultures was evidence for the foundational idea that learning and knowledge are always situated in the context of learners' lives [1]. Such participatory spaces create communities of practice where learning occurs through participation in the community. In this model, learning CS is learning how to participate in computing communities of practice, a necessarily social process which involves doing things within a community of practice, experiencing the actions and reactions of others within that community of practice, developing a sense of belonging to the community, and ultimately becoming someone within that community of practice [23]. A student's identity is formed by what they see as possible forms of participation in these computing communities of practice as well as what other members of the community accept as legitimate participation [9].

## 2.2 Material intelligence

Because computational artifacts are integrated in nearly every part of our lives, computing communities of practice exist across formal education spaces like CS classrooms as well as in everyday spaces like online forums for video games or social media spaces where students create and share memes. Most students are involved in many, diverse computing communities of practice simultaneously [2]. However, the forms of participation a student sees as possible in these distinct spaces are often different, and a student's computational identity is also affected by how they navigate between these different communities of practice, positioning themselves in different ways based on their unique experience of each community of practice [22]. Nevertheless, the computational ideas and often the practices themselves which mediate participation in these communities are shared. This ability to leverage representational media for participation in communities of practice is what diSessa [6] refers to as "material intelligence." As learners develop their material intelligence with these ideas and practices, new properties of computing communities of practice and potentially new ways to participate in them become available to students as their knowledge is reformulated through "restructurations" [24]. These restructurations affect students' computational identities as they apply the shifts in their knowledge across their many communities of practice and, as a result, as they reposition themselves within those communities of practice.

## 2.3 Liberatory pedagogy

This positioning across communities of practice is how students author identities relative to a specific discipline [10]. A student's capacity for identity authorship relies on their current positioning as well as what they see as their possible future positioning–their trajectory of participation within their communities of practice [7]. Many trajectories of participation exist and while some move students deeper into a community of practice, others produce distance from a community of practice. Rather than being fixed determinations of a learner's identity, these trajectories of participation provide resources and constraints for a learner's identity authorship. The space for authorship defined by these resources and constraints is where we locate agency in the context of a discipline [21]. Cultivating this space for agentic, self-directed participation in computing communities of practice is pivotal for promoting education as an emancipatory, political act [3]. By supporting students in developing intentional, empowered computational identities based on their participation in authentic computing communities of practice, students can leverage their CS education in "reading the world" [8].

Understanding students' experiences with CS education through their developing computational identities is aligned with Papert's vision of computing as an inherently productive, social practice [12]. This definition of a computational identity helps clarify our goal for CS education as supporting students in developing empowered computational identities which allow them to determine for themselves *how* and *whether* to participate in the many computer cultures that surround them. With this goal in mind, we seek to understand how students' computational identities develop as they learn CS.

## 3 METHODS

This study developed out of a research-practice partnership [5] between a US research university and a private bilingual K-12 school in Hong Kong, where 28 ninth graders participated in a two-year curriculum. In this cohort, 13 consented to participate in this research. The central goal of the course is to "create a rich, diverse community of people making things with code, through which they can develop personal relationships with powerful ideas" [16]. The course is composed of six thematic units, each focused on a domain of computational media production such as art, games, data science, and web development. In each unit, students completed an open-ended project that applied skills and concepts from that unit. At the end of the two-year program, three researchers who were also teachers conducted a semi-structured exit interview with each student. In hour-long interviews, we asked students to reflect on formative moments with technology throughout their lives as well as changes in their relationship to their computers, the internet, and the software tools used in the course (such as Git and Python) over the course of the two years. Interviews were conducted in English, recorded, and professionally transcribed. To analyze the data for this paper, three of the authors used a phenomenographic methodology [14], conducting a thematic analysis [4] to inductively identify themes in students' reflections about the computational world and their place in it. One author then re-coded each episode of reflection on the computational world to identify instances of "re-seeing" the computational world. Within these instances, the author then used a grounded approach to develop the three categories of re-seeing presented in the findings. Students' names have been changed for anonymity.

## 4 FINDINGS

Though they had a wealth of experience with technologies from gaming PCs to social media to learning management systems prior to the course, students frequently self-reported a change in their interactions with everyday technologies during their exit interviews after the course. Reflecting on her experience watching a music video on a Virtual Reality platform, one student, Tina, said:

> "It made me think more about what I was using and what I was doing. And like the interaction between person and digital device, that kind of relationship. . . Like, "oh," if I click on this button, there's a whole bunch of code behind the color of this button and the depth to make it look like a button. And this button links with some other page. And it's like a whole process of coding and effort for me to just click on a button to get to a new page."

In total, we found that ten of the thirteen interviewed students (77%) discussed an experience where they used technology differently or felt differently about technology compared to their experiences before the course. Like Tina, these students described a realization of the many layers through which their interactions with technology flowed. In analyzing students' reports of re-seeing the computational world, we found three common themes in their experiences: (1) respect for the work of creating technology, (2) curiosity about how technologies worked, and (3) an impetus to do more with their use of technology.

### 4.1 Respect

Some students shared that the course had given them an increased respect for people who create technology, like software engineers and web developers. In describing how learning CS changed the way he saw technology, Teddy talked about respect explicitly:

> "It has made me gain more respect for software developers and how they maintain websites. For example, Facebook. There's just so much stuff you need to do to have it functioning all the time, bug fixes and other parts."

Other students shared similar sentiments without explicitly using the word "respect" when they shared comments about realizing how much work goes into technology development. Tina provided an example of this kind of respect when talking about how she began to perceive digital experiences differently:

> "I think it was during the games unit or something afterwards, like whenever I played a game on my phone or something, I'll be like, oh, wow, this must have taken some time to do. I don't know how they did it, but it must have taken some time for them to make this work."

This kind of respect was a common experience for students with episodes from five different students (38%) coded as respect, and many students connected their re-seeing experiences to respect for computer scientists explicitly or implicitly with comments like Tina's. Some students tied this sense of respect to their developing ideas of what a computer scientist does, from specific activities like building websites or fixing bugs to more generalized ones like "solving problems with code" or "[increasing] the quality of life of other people." A few students also drew from these notions to help determine whether they themselves could be computer scientists, either by seeing these activities as similar to the kinds of things they had done or as something much more complicated than what they could do.

### 4.2 Curiosity

Beyond being impressed that someone could make a complicated technology, some students wanted to know how they actually made it. Five students (38%) expressed a sense of curiosity in their re-seeing experiences. Rebecca described this experience when questioning how a high-level programming language used to create video games implements more foundational layers like graphical rendering:

> "Part of it would be just called, I guess, newfound respect. . . but it's more of curiosity. Just how did they

> do that? Just going up the layer, how does the very top guy just make that happen?"

Rebecca connected this curiosity about graphical rendering to her experience making a drawing in our computational art unit, wondering how her code gets translated to pixels on a computer screen. Other students also connected their curiosity to other projects in the class. Liam, for example, sought out resources to understand what was possible beyond what was taught in class and described his experience of reading documentation for web frameworks during the web apps unit:

> "You're never taught it, but you could read a documentation and figure out how to solve the problems yourself. You have a lot more options if you read the documentation and be like, 'I didn't know you could actually do this."'

Further, some students' curiosity also sparked a burgeoning critical perspective on computing. When confronted with the complexity of the internet, Ren began to question the security of a system he had previously trusted:

> "I guess it made me more wary of logging into, or basically using the internet, I guess. It made me question how secure some websites are or how safe my data is on the internet. I guess, especially in the web apps unit, since you're able to develop your own website, it helps me understand how fragile the system is, I guess. Because there's not a lot of security for [my] website and it helped me understand that a lot of other websites might be the same as well."

Ren saw new ways data is collected, stored, and used on the internet, and, subsequently, began to wonder about the risks associated with those practices. Ren's questioning of the security of websites is an indication of how his understanding of the computational world is changing. While he previously saw websites as obscure but well-protected systems, his experience of creating a website himself has helped him realize the fragility of these systems. Ren's comments offer another example of the way students' experiences of curiosity about technology were often driven by their experiences creating a computational artifact.

### 4.3 Doing more

While Ren's quotations offer an example of burgeoning critical perspective, other students described concrete changes in the way they use technology. Along with their shifting perspectives, 6 students (46%) were activated to do something new because of what they now saw. Liam describes how his deepened understanding of data collection practices changed the way he uses the internet:

> Liam: "I didn't realize that these big companies actually used a lot of the data, and they kept it. A lot of the time it doesn't make sense why they have it; they just have it. I didn't realize that you could just turn these functions off, but it's not taught to you."
>
> Interviewer: "Did that change any of your habits?
>
> Liam: "Yes, I actually did change quite a few of my options to be non-tracking, especially a site that didn't need it in the first place."

Another student, Noa, experienced the impetus to do more in a more holistic way. During the data science unit, Noa chose to find her own datasets to answer a question about her experiences as a competitive athlete in a water-based sport. Reflecting on this experience and how she developed over the two years of the course, Noa said:

> "The data science [project] actually helped me develop as a person, because that actually helped me understand data much better. I feel like I'm a very environmental aware person. I really care about environment. I really want to help the environment improve. And by doing data structures, led me on to give me opportunity to [write a conference publication] and many other projects developed after that... I think a lot of it was character development, me growing as a person and as I mature, I feel I'm being more aware of the people around me, and the events happening around me... I'm actually doing more service. I want to do more for the world."

The data science project was a critical moment in Noa's trajectory as a computer scientist. Noa's experiences in the course allowed her to see the way computation could be useful for answering questions and solving problems that were relevant to her life. Once she had glimpsed this utility, she became activated to continue utilizing her computational perspective and skills throughout her life and explicitly connected her experiences creating computational artifacts to her growth and development as a person.

## 5 DISCUSSION AND CONCLUSION

To understand what our findings reveal about how students experience their computational identities changing as they learn CS, we put the three categories of re-seeing identified in the last section in dialogue with one of the three themes of Constructionism used to develop our definition of computational identity.

In their descriptions of the respect they developed for computer scientists, students drew from their experiences within computer cultures from social media platforms to mobile gaming. In discussing how their CS education affected these experiences, our students described looking at a computational artifact, imagining the people who were behind the creation of that artifact, and then considering themselves in relation to those people. In doing so, our students demonstrated their growing awareness for computing communities of practice and the positioning they do relative to those communities. While this positioning is always happening, our students' comments indicate that their CS education helped to make this positioning a more conscious act. Even when they did not have inbound trajectories toward computer science, students "re-seeing" of technology reinforced their peripheral participation as informed observers.

Students' discussion of curiosity about their computational worlds demonstrates how they applied the material intelligence they developed during their CS education. In Rebecca's curiosity about computer graphics in gaming, she applied her understanding of abstraction within computational systems to begin breaking them down into discrete "layers" to refine her questions about how things work. Rebecca explicitly connected this material intelligence with her experience creating a project in our course. Ren's comments about his growing wariness of the internet demonstrated how this developing material intelligence led students to re-evaluate and reposition themselves within computing communities of practice.

The experiences of "doing more" students described have clear connections to liberatory pedagogy. From a refusal of personal data collection to a transformative desire to improve the world with computing, students discussed how their experiences in the course changed the ways they interact with the world. In doing so, our students demonstrated how when more resources (like knowledge of data tracking practices or data science skills) became available to them, they felt more empowered to author their computational identities, making more intentional decisions about who they would be in the computer cultures in which they participated.

Taken together, our findings demonstrate that there is no "one-size-fits-all" model for computational identity and that students' computational identities can be meaningfully authored even if those identities do not lead them to normative endpoints like formal study of CS. Instead, each student has a nuanced relationship to computing that changes over time as students re-see their experiences with computation with respect, curiosity, and an impetus to do more. These re-seeing experiences are key moments in students' computational identity authorship as they prompt them to consciously position themselves in computing communities of practice. Further, as students navigate these moments of identity authorship, they readily draw on their experiences creating personally meaningful computational artifacts and on their positioning within the computing community of practice in our course.

As CS educators, we should remain humble, remembering that even if the time we spend with students is impactful, it is a small part of their lives, and even a small part of their computational identities. Our goal should be to contribute resources and provide a supportive context for the growth of youth computational identities, but we should remember that they are on long trajectories, woven into more contexts than any one class. In future work, we plan to investigate how specific features of our course impact students' developing computational identities. In doing so, we hope to continue exploring what Constructionism can offer CS education while using CS education as a site to develop Constructionist philosophy and pedagogical practice.

## REFERENCES


[1] Ackermann, E. (2001). Piaget's constructivism, Papert's Constructionism: What's the difference. *Future of learning group*, 5(3), 438.

[2] Anderson, M., & Jiang, J. (2018). *Teens' Social Media Habits and Experiences.* Pew Research Center.

[3] Blikstein, P. (2008). Travels in Troy with Freire: Technology as an Agent of Emancipation. In C. A. Torres & P. Noguera (Eds.), *Social Justice Education for Teachers* (pp. 205–235).

[4] Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), 77–101.

[5] Coburn, C. E., & Penuel, W. R. (2016). Research–Practice Partnerships in Education: Outcomes, Dynamics, and Open Questions. *Educational Researcher*, 45(1), 48–54.

[6] diSessa, A. A. (2001). *Changing minds: Computers, learning, and literacy.* Mit Press.

[7] Dreier, O. (1999). Personal Trajectories of Participation across Contexts of Social Practice. *Outlines. Critical Practice Studies*, 1(1), 5–32.

[8] Freire, P., & Macedo, D. (1988). *Reading the Word, Reading the World.*

[9] Gee, J. P. (2001). Identity as an analytic lens for educational research. *Review of research in education*, 25(1), 99-125.

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

[11] Kafai, Y. B. (2016). From computational thinking to computational participation in K-12 education. *Communications of the ACM*, 59(8), 26-27.

[12] Kafai, Y. B., & Burke, Q. (2014). Mindstorms 2.0: Children, Programming, and Computational Participation. *Constructionism.*

[13] Kafai, Y., Proctor, C., & Lui, D. (2020). From Theory Bias to Theory Dialogue: Embracing Cognitive, Situated, and Critical Framings of Computational Thinking in K-12 CS Education. *ACM Inroads*, 10.

[14] Kinnunen, P., & Simon, B. (2012). Phenomenography and grounded theory as research methods in computing education research field. *Computer Science Education*, 22(2), 199–218.

[15] Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas.* Basic Books.

[16] Proctor, C., Han, J., Wolf, J., Ng, K., & Blikstein, P. (2020). Recovering Constructionism in computer science: Design of a ninth-grade introductory computer science course. In B. Tangney, J. Rowan Byrne, & C. Girvan (Eds.) *Proceedings of the 2020 Constructionism Conference.* (pp. 473-481). Dublin, Ireland: University of Dublin.

[17] Schulte, C., & Budde, L. (2018). A Framework for Computing Education: Hybrid Interaction System: The need for a bigger picture in computing education. In *Proceedings of the 18th Koli Calling International Conference on Computing Education Research* (pp. 1-10).

[18] Shaw, M. S., Ji, G., Zhang, Y., & Kafai, Y. B. (2021). Promoting socio-political identification with computer science: How high school youth restory their identities through electronic textile quilts. In *2021 Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, 1–8.

[19] Silver, J. J. S. (2014). *Lens x block: World as construction kit* (Doctoral dissertation, MIT).

[20] 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.

[21] Tolman, C. (1994). *Psychology, Society and Subjectivity (1st ed.).* Routledge.

[22] Van Langenhove, L., & Harré, R. (1999). Introducing positioning theory. *Positioning theory: Moral contexts of* intentional action, 14-31.

[23] Wenger, E. (1999). *Communities of practice: Learning, meaning and identity.* Cambridge University Press.

[24] Wilensky, U., & Papert, S. (2010). Restructurations: Reformulations of knowledge disciplines through new representational forms. *Constructionism.*

[25] Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.

[26] Wolf, J., Han, J., Proctor, C., Brown, E., Pang, J., & Blikstein, P. (2023). "Growing as a person": Authoring Identity Across Formal CS Education and Everyday Computing Contexts. In *Proceedings of the 17th International Conference of the Learning Sciences - ICLS 2023.* Montreal, Canada: ISLS.