Technology, Knowledge and Learning
https://doi.org/10.1007/s10758-022-09606-w

ORIGINAL RESEARCH

The State of Computational Thinking in Libraries

David Weintrop[1,2] · Mega Subramaniam[2] · Shandra Morehouse[3] · Nitzan Koren[2]

Accepted: 21 April 2022

Abstract

Computational thinking is an essential 21st-century skill that all youth should develop in order to navigate and succeed in an increasingly computational world. For all youth to have hands-on opportunities to develop essential computational thinking skills, libraries and informal learning environments play a critical role. This is especially true for youth who attend schools where computational learning opportunities are limited or altogether absent. This article presents an analysis of responses from 59 library staff members to the following questions: What is computational thinking? How is computational thinking being facilitated in libraries? And, what are the goals of the computational thinking programs being offered by libraries? The analysis reveals the multifaceted ways that library staff conceptualize computational thinking and the range of ways computational thinking is being integrated into library programs. This work advances our understanding of the current state of computational thinking in libraries. In doing so, we seek to guide library staff on ways to support the computational thinking learning that is currently happening and create new ways to help bring the powerful ideas of computing to broader audiences.

Keywords Computational thinking · Libraries · Library programming

✉ David Weintrop
weintrop@umd.edu

Mega Subramaniam
mmsubram@umd.edu

Shandra Morehouse
shandra@umd.edu

Nitzan Koren
nkoren@umd.edu

[1] College of Education, University of Maryland, College Park, MD, USA

[2] College of Information Studies, University of Maryland, College Park, MD, USA

[3] Library of Congress, DC, Washington, USA

Published online: 16 May 2022

1 Introduction

Computing, and the technologies it enables, are playing an increasingly prominent role in society. Given this growth, it is essential that all youth have the opportunity to develop foundational understandings of what computing is, what can be done with it, and how it impacts their lives. Increasingly, these topics are captured under the term *computational thinking* (CT) (Wing, 2006). Broadly, CT describes the skills, concepts, and practices associated with effectively using computational tools to solve problems (Curzon et al., 2019; Grover & Pea, 2013; Shute et al., 2017). More concretely, CT includes concepts such as abstraction, decomposition, and conditional and iterative logic along with practices such as problem decomposition and debugging, however, the specifics of what does, and does not, constitute CT remains an active debate (Denning, 2017) (see Sect. 2.1 for an extended review of CT). While schools play an important role in providing CT learning opportunities, there are aspects of CT, such as tinkering, exploration, and using computing for personally meaningful pursuits, that are not well suited for the disciplinarily organized and regimented structure of contemporary K-12 education. Further, not all schools and classrooms are well-equipped to teach CT for a variety of reasons including lack of resources (e.g. computers, curricula), lack of time during the school day, and lack of experienced teachers to lead computing classes (Angeli et al., 2016; Weintrop, Beheshti, et al., 2016; Yadav et al., 2017). As such, informal learning environments, including libraries, museums, camps, and afterschool programs can provide a space for youth to learn foundational CT concepts and skills in a way that complements the strengths of formal education (Braun & Visser, 2017; Pinkard et al., 2020; Subramaniam et al., 2019).

Given the importance of CT and the essential role that libraries play in youth learning (National Academies of Sciences, Engineering, and Medicine, 2021), it is not surprising to see a growing number of CT-related tools and programs[1] being included in the services offered by libraries (Braun & Visser, 2017; Lee & Phillips, 2018; Subramaniam et al., 2022; Weintrop et al., 2021). It is now commonplace to find robotics kits, 3D printers, digital fabrication tools, and banks of computers, tablets, and laptops in libraries alongside books and periodicals. This growth in the presence of technologies associated with CT in libraries has happened quickly and takes different forms across different libraries. Driven by different visions, opportunities, community needs, and resources, there is relatively little systematicity as to how libraries have brought CT programming to the communities they serve.

The goal of this work is to try and understand and characterize the current landscape of CT in public libraries.[2] Towards this end, this paper pursues a set of questions designed to shed light on the form and substance of CT in libraries as it currently exists. Specifically, this work seeks to answer the following questions:

> How do library staff conceptualize computational thinking? What concepts and skills do they think it includes?

[1] We use the terms "programming" and "programs" in this paper to describe structured activities offered in libraries (e.g., "library programs" or "CT programming"). This is a common vernacular in library scholarship.

[2] The phrase "CT in public libraries" is intended to be inclusive of all library programs and activities offered by public libraries at the library, online, and in other community spaces.

How is computational thinking being introduced into public libraries? Who is leading and facilitating computational thinking programming? What resources and tools are being used?

What are the stated goals for the computational thinking programming being offered by public libraries?

To answer these questions, we conducted interviews and focus groups with 59 public library staff from across the United States who are leading CT programming in their libraries. In analyzing these interviews, we gain a sense of commonalities and differences in the ways CT programming is being conceptualized and offered in libraries. We also learn the roles that public libraries are playing in introducing youth to CT and shed light on gaps that exist and the associated opportunities for libraries to better serve their communities.

This paper begins with a review of literature focusing on CT broadly, CT in informal contexts specifically, and a review of prior research on the role of technology in libraries. We then describe the methods used to answer our stated research questions before presenting our findings. The paper concludes with a discussion of the findings and implications for this work as it relates to both libraries as well as those whose work seeks to improve and support them. Collectively, this work advances our understanding of the current state of CT in libraries and helps provide a roadmap for the important work that lies ahead to help libraries achieve their full potential as venues to help youth succeed in an increasingly computational world.

2 Prior Work

2.1 Computational Thinking

The idea that computers can serve as powerful tools for thinking and learning was initially developed and explored by Papert et al. (). Through the development of the Logo language, Papert et al. investigated how programming can serve as a context for mathematical learning as well as the formation of productive problem-solving strategies and critical thinking skills (Harel & Papert, 1990; Papert, 1972). In reflecting on this work, Papert concluded: "computer presence can contribute to mental processes not only instrumentally but in more essential, conceptual ways, influencing how people think even when they are far removed from physical contact with a computer" (1980, p. 4) which serves as an important antecedent to the contemporary CT discussion. This work shaped decades of research on the design of computational learning environments (e.g. Abelson & diSessa, 1986; Kay & Goldberg, 1977; Resnick et al., 2009; Wilensky, 1999) and lead to the emergence of CT as a term to capture the variety of ways computing can inform how we think and solve problems (Grover & Pea, 2013).

The contemporary discussion of CT begins with Wing's, 2006 article in which she argues: "to reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability" (2006, p. 33). Since its publication, significant effort has been put towards understanding the nature of CT and exploring ways to create learning opportunities for youth to engage with it (Curzon et al., 2019; Grover & Pea, 2013). In response to the rise of scholarship related to CT, the National Research Council organized a pair of meetings to discuss the nature of CT (2010) and its pedagogical aspects (2011). The result of these meetings was a consensus around the importance of CT as a set of skills all learners should develop but less agreement on exactly what CT is and where the boundaries of the construct lie. A recent review of CT literature by Shute et al. (2017) seeking to identify the major characteristics and components of CT concluded that CT consists of the following facets: decomposition, abstraction, algorithms, debugging, iteration, and generalization.

While this consensus definition provides a broad sense of what CT is, other researchers have worked to define CT in more constrained contexts. By tailoring a definition of CT to a narrower context, it becomes easier to identify specific CT concepts and skills and how they apply within the specified context. For example, while focusing on the Scratch programming environment, Brennan and Resnick (2012) proposed a framework for CT that includes CT concepts (e.g. sequences, parallelism, conditionals), practices (e.g. abstracting and modularizing, testing and debugging, remixing, and reusing), and perspectives (e.g. expressing, connecting, questioning). A second example can be seen in the development of the CT in math and sciences practices taxonomy, which operationalizes CT in the context of math and science into four sets of practices: data practices, modeling and simulation practices, computational problem-solving practices, and systems thinking practices (Weintrop, Beheshti, et al., 2016). With these examples, we begin to see how CT can take on different forms in different contexts. Given the lack of a singular definition and the unique affordances and constraints as libraries as venues of learning, it is not yet well understood how libraries and library staff operationalize this emerging concept of growing importance in their own programming. This work addresses this gap. In the next section, we being to review the work done to date in this space.

2.2 Computational Thinking in Libraries and Informal Spaces

While most research on CT has focused on learning in formal contexts, there is a growing body of research exploring CT learning outside of the classroom. This includes libraries, museums, makerspaces, afterschool programs, and a growing number of toys and video games designed to engage learners with CT through play. Opportunities for youth to engage with CT outside of schools are essential to help situate CT skills and concepts within the values of a community in equitable and accessible ways (Pinkard et al., 2020). Here we review research on CT in libraries first, then briefly review work on CT in additional informal contexts as a means to provide a sense of the nature of CT outside of formal learning environments and situate our work on the state of CT in libraries.

Given their ubiquity and their roles as community resources, public libraries are important contexts for computational learning and exploration as well as venues for promoting digital literacy in youth (ALA, 2020; Taylor et al., 2018). A growing number of library-based initiatives are working to introduce computing and CT as part of their library programming (Martin, 2017; Subramaniam et al., 2019). For example, the American Library Association (ALA)/Google initiative called *Libraries Ready to Code*, supported library staff to bring CT and CT-related activity (e.g. programming, robotics) into their libraries (Braun & Visser, 2017). CT programming in libraries uses various virtual tools such as Scratch and Minecraft as well as physical computing kits like Ozobots, Lego Mindstorms, and Sphero robots (Guidara, 2018; Prato, 2017; Wing & Meyers, 2014). Digital fabrication, 3D printers, paper circuits, and e-textiles are also commonly used in libraries as a way to bring CT to young library patrons (Einarsson & Hertzum, 2020; Lee & Recker, 2018; Prato, 2017). To date, the focus of these initiatives has largely been tool and technology based, rather than a deeper consideration of conceptual underpinnings or learning outcomes. Further, little empirical work has sought to map out this landscape across initiatives, instead, focusing on specific programs or technologies.

Alongside libraries, museums serve as a second informal institution where youth can engage with CT. A growing number of museum exhibits are seeking to make computing or ideas related to CT accessible to museum visitors (Horn, Weintrop, et al., 2014; Mesiti et al., 2019; Metcalf & Anderson, 2020). Like with libraries, CT exhibits in museums use a variety of technologies and are situated in varying contexts. For example, Horn et al. have developed CT museum exhibits focused explicitly on programming in the form of the Robot Park exhibit (Horn et al., 2009) and CT in the context of natural selection with the Frog Pond exhibit (Horn, Brady, et al., 2014). Another context for situating CT in museum settings can be seen in the use of engineering activities focused on construction, design, and testing as a means to engage young museum visitors in authentic CT practices (Ehsan et al., 2020).

Community centers, after-school programs, summer camps, and organizations such as 4-H, Boys and Girls Clubs, all provide additional contexts where CT learning occurs. For example, Pinkard et al. organized the Digital Youth Divas, an out-of-school program that supports non-dominant middle school girls in developing digital artifacts, building STEM interests, and affect change in their local communities (Pinkard et al., 2017, 2020). A final context beyond schools where CT learning is occurring is the home. A growing number of toys and games focused on developing learners' CT skills (Hamilton et al., 2018; Yu & Roque, 2019). Along with toys, research has also found that productive CT practices can emerge through playing board games (Berland & Lee, 2011; Tsarava et al., 2018) and as part of designing and/or playing video games (Holbert & Wilensky, 2011; Lee et al., 2014; Weintrop, Holbert, et al., 2016). Collectively, this review shows that research has been conducted on CT in libraries and other inform spaces, but there is little insight into the prevalence of these programs at scale, a gap this research seeks to address.

2.3 Technology Programs in Libraries

Libraries are often a hub for community activity and provide vital community resources like access to technology (Subramaniam et al., 2018; Thompson et al., 2014). Thus, libraries provide opportunities for users of all ages to learn more about technologies, how to use them, and the literacies associated with these technologies (Thompson et al., 2014). Library spaces are especially important for providing access and education on technology to the youth in their communities (Lee, 2019; Tzou et al., 2019), especially youth from populations historically excluded in computing such as youth from racial minority backgrounds and youth that live in rural areas (Davis et al., 2018). Libraries provide programs that allow youth to interact with technologies, like computers and robotics kits, and learn various skills like coding, game making, and media creation (Braun et al., 2017; Davis et al., 2018; Subramaniam et al., 2018). Technology programs in libraries provide opportunities for those who would not normally have access to different technologies (Subramaniam et al., 2018; Vickery, 2014) and play an especially important role in rural communities where access to high-speed broadband is limited (Real & Rose, 2017).

Technologies such as robots, multimedia production tools, and digital fabrication tools also help libraries to offer programming that interests youth. One way libraries have done this is by using the connected learning framework (Ito et al., 2013) to create programs that combine youth's interests with learning new tech skills (Hoffman et al., 2016). The connected learning framework focuses on creating learning experiences that are driven primarily by the youth's interests (Ito et al., 2020; Pinkard et al., 2017). By drawing on youth's existing interests, libraries can create personally relevant and engaging learning experiences (Barron, 2006). Another way libraries use technology to reach youth is through the introduction of Makerspaces and Learning Labs (Dresang, 2013; Koh & Abbas, 2015; Koh et al., 2018). Many libraries have Makerspaces (Garmer, 2014), further serving as a vital access point for Making practices and experiences for youth in many communities (Tzou et al., 2019). Makerspaces, and the Making culture more broadly, serve as important communities where CT learning occurs outside of classrooms (Hadad et al., 2020; Peppler et al., 2016).

Another increasingly common form of library programming related to CT is activities to teach computer programming. Many of these programs use Scratch, a web-based programming platform that allows for drag and drop programming that also has several easy-to-follow curricula for library staff to use (Martin, 2017; Prato, 2017). Libraries also teach programming with text-based languages like HTML, Python, and Java (Prato, 2017; Vickery, 2014). Unplugged activities are also a popular way for library staff to teach youth CT concepts, and are especially useful for smaller libraries that have fewer resources (Lee & Recker, 2018; Prato, 2017).

Across these various types of technology-infused library programs, one central consideration is the prior knowledge and skills library staff need to support such programming. Many library staff are new to the concepts involved and often rely on easy-to-use tools and prescriptive activities designed for use in libraries (Braun & Visser, 2017). In response to this, there are a growing number of programs designed to help support libraries and library staff bring technology into their buildings, such as Google's CS First, Girls Who Code, and the Libraries Ready to Code (RtC) initiative. For example, the RtC program provides resources on its website for library staff to use and adopt as well as grants for library training programs (Taylor et al., 2018). This review of the literature shows that technology-rich programming is already present in libraries, meaning that CT-related content is already being introduced (or there is great opportunity to introduce it). However, little is known about how library staff conceptualize CT in their programming or the ways that CT concepts and skills manifest in the programming that is currently being offered. The research presented in this paper contributes to the literature by investigating this gap and advancing our understanding of the current state of CT in public libraries.

3 Methods and Participants

To investigate the current landscape of CT in libraries, we recruited youth-serving library staff working in public libraries across the US to participate in semi-structured interviews. Each interview lasted 30–60 min and was conducted virtually and audio recorded. Another 15 youth-serving library staff took part in 4 focus groups—an in-person focus group at the Young Adult Library Services Association's (YALSA) 2019 Symposium and three virtual focus groups. The focus groups lasted 60–90 min and were audio-recorded. Participants were offered a $25 Amazon e-gift card for participating.

The decision to use semi-structured interviews and focus groups was motivated by the desire for extended discussion and provide participants with a space to articulate their ideas and share details of their experiences in hopes of revealing subtle nuances in their thinking

Table 1 The five focal codes from our codebook used in this work

Code	Definition
CT Motivations, Goals, and Outcomes	Responses attending to motivations for CT programs, the goals they hope to accomplish, and the outcomes they would like to see in the youth participants
CT Definition	Responses to our interview question about familiarity with CT and how they define it
CT Resources	Any mention of tools, programs, curricula, or computing languages used along with how they have used it, and the reasons for using it
CT Experience, Exposure, and Training	Responses attending to how they were introduced to CT and the ways they have advanced their understanding of the concept
CT Programs	Any description of what youth are doing when engaging in CT in and via the library (e.g., a specific activity, a program, etc.)

and motivation around CT to answer the research questions being pursued. The decision to use focus groups in addition to the semi-structured interviews was based on prior experiences of the authors with this methodology (XXXX, 2018; XXXX, 2021) and the fact that focus groups allow participants to learn from peers. This context can spur participants to reflect on their practices and ideas, thus providing additional, and at times deeper, insight into their views (Morgan, 1996). Further, during focus groups, it is common for a single idea to be brought up and then to hear other participants build on how the idea manifests in different contexts, which was particularly generative for this work.

For the interviews and in-person focus group, we recruited via social media, our project website, YALSA's e-newsletter, and via professional networks of the research team and project partners. To ensure that we recruited library staff familiar with CT, we included the following language in all our recruitment materials—"If you have hosted a computational thinking program at your institution or are planning to offer a computational thinking program in the near future, we would like to do a virtual interview with you to learn more about your experiences in computational thinking programming". Participants were asked to self-report the populations that they serve with 25.5% indicating that they serve urban populations, 40% serving rural populations, and 34.5% serving suburban populations. All regions of the US are represented (20.4% Northeast US, 32.2% Midwest US, 23.7% West US, and 23.7% South US). Their work experience ranged from 1 to 40 years with an average of 8.8 years of experience.

The interview and focus group used a protocol that covered topics including the library staff's experience with CT, their own definition and understanding of CT, the programs they ran at their library, the CT-related tools/technologies they have used, their motivations for offering CT programs, and the challenges that they face. The interview protocol can be found in Appendix.

We employed thematic analysis (Boyatzis, 1998) to create a coding scheme aligned with our research questions. In the first stage of coding, each researcher separately coded a single interview transcript, and we came together to discuss the codes applied and developed an initial codebook. We then used this codebook to code an additional two transcripts individually, then met to discuss the results of our coding and finalized our codebook. The final codebook, containing 16 codes, was then applied to all transcripts by one member of the team with an additional team member reviewing and modifying the coded transcripts as needed. The research team conducted regular collaborative discussions throughout the iterative analysis process to ensure consistency and accuracy (Smagorinsky, 2008). To address the research questions being pursued in this paper, we specifically focus on five codes that emerged in our analysis, which are introduced and defined in Table 1. The other 11 codes were related to topics covered in the interview (e.g., assessment strategies, challenges of running CT programs) but not directly related to the specific questions being answered here.

4 Findings

4.1 What is Computational Thinking According to Library Staff?

To answer the question of what CT is according to library staff, we included the following question in our interview protocol: *Are you familiar with the term computational thinking? If so, how do you define it?* From the responses, we identified 13 distinct components of CT that capture the most frequently cited aspects of CT. The 13 components of CT along with the number of participants who included that idea as part of their response to the interview question and a sample quote demonstrating each component are presented in Table 2. The responses we received to this question often included multiple components of CT. On average, the responses included 2.2 components of CT per response (*SD* 1.1) with the largest number of components identified in a single response being 6.

The responses we received to our definitional CT question highlight the diversity of ways that library staff make sense of the concept. While there is a common theme across the components of CT in how many responses linked CT to systematic approaches to problem-solving and using computing to help solve problems, the specifics that library staff choose to focus on vary widely. One noteworthy aspect of many of these responses is how they differ from the types of responses heard when interviewing classroom teachers as identified by prior research. While both classroom teachers and library staff most often cited problem-solving as a central component of CT, classroom teachers were more likely to focus on technological aspects of CT (Garvin et al., 2019), whereas library staff often focused on more library and communities-oriented instantiations of the concepts or general activities that apply broadly to the lives of the youth their library programming is serving, as can be seen in Molly's definition linking algorithms to story time in Table 1.

Another interesting aspect of how library staff define CT is in how their conceptualizations of CT and how they frame it to youth are tied to social and societal issues. For example, when asked how she defined CT, Cait (Urban, West) responded with the idea of pattern recognition saying, "you got the pattern recognition, you know, looking for similarities" and then quickly transitions to social issues, continuing, "we gave them social issues that happen in our community and we had them apply computational thinking to that specific social issue." She then went on to explain how she worked with kids to identify patterns in challenges that homeless youth face when starting school, such as a lack of school supplies and then develop apps to help solve them.

We also coded responses that equated CT to "thinking like a computer". We included this code as it is a common misinterpretation of CT and in some ways antithetical to the goals of the CT movement. A central component of CT is the knowledge and skills associated with taking advantage of computers, not being a computer (Grover & Pea, 2013). In this way, the goal of CT is to let the computer "think like a computer" and for the learner to

Table 2 Components of the ways that library staff define CT

Components of CT	Frequency	Sample quote
Problem-solving	31	I define it as the process that we go through to figure out how to solve problems, however, you define any of those terms. (Paige[a], Urban Northeast)
Algorithms	23	We do computational thinking in a form of directions, following steps, I mean, our story time can do computational thinking… (Molly, Suburban Southeast)
A way of thinking or logical thinking	17	To me, computational thinking is learning and thinking in a step by step logical process. And adding on to those steps. (Patricia, Rural West)
Problem decomposition	14	It's a process to be able to break down problems into smaller pieces to get it done, to figure it out. (Stella, Urban Southeast)
Related to programming or coding	12	A set of skills that allow problem-solving, particularly in terms of computers and coding. (Bailey, Rural Midwest)
Link to disciplinary thinking	7	When you really look into it, computational thinking is just…implementing the scientific method. (Michelle, Rural Southeast)
Thinking "like a computer"	7	It's the process of almost thinking like a computer. (Jean, Suburban Northeast)
Creativity	4	I would define it as, basically, I guess, a logic-based approach to thinking and problem solving, but that emphasizes resourcefulness and creativity instead of rote memorization. (Tyler, Suburban Northeast)
Pattern Recognition	4	…breaking it down into smaller parts and recognizing a pattern that creates that and being able to take those parts that make it all work (Brittany, Rural Southwest)
Abstraction	4	It's a lot of what computer programmers do. It is using and how to understand algorithms and abstraction (Tessa, Suburban Midwest)
Debugging	2	I would describe that as discovery through trial and error. (Lila, Suburban Northeast)
Automation/Iteration	2	It's breaking the problem down into different steps and perspectives that help us to find points of automation and find points of repetition (James, Urban Southwest)
No Answer/Not familiar with CT	4	I am [familiar with it], I don't quite understand it though. I'm sure that I do it, I just don't know exactly what it is. (Tara, Suburban Midwest)

[a]All names are pseudonyms and each quote is attributed to a person and includes their geographic region and the type of community where their library is located (urban, suburban, rural)

take advantage of what the computer can do. This misinterpretation is widespread among those new to CT.

A final noteworthy finding from this analysis was both the breadth of how library staff conceptualized CT but also a recognition that the concept and their understanding of it can change over time. As Molly (Suburban, Southeast) said in her response "My definitions have evolved over the past three years because my programs have changed with the times and stuff too. But originally, I was just thinking computational thinking or computer science, that means we sit down at a laptop or some form of computer and they're going to have to code…Now, I understand that that's not necessarily what that means. Now, we do computational thinking in the form of directions, following steps, I mean, our story time can do computational thinking, you know what I mean?" In this quote, we see one library staff's growth in her thinking as well as how that growth resulted in changing how and where CT happened in her library.

4.2 How is Computational Thinking Being Facilitated in Libraries?

To answer our second research question exploring how CT is making its way into libraries, we investigate the tools and resources being used for CT programming in libraries, the library staff who are running CT programming, and, finally, share findings around structural aspects of CT programming in libraries, including when and how often CT programs are offered.

4.2.1 What Resources and Tools are Being Used to Teach Computational Thinking in Libraries?

We began our analysis on the tools and resources being used in library-based CT programming by pulling out every mention of a technology or a resource from our interview data and grouping them into categories to get a general understanding of the landscape of how CT is being offered. Our analysis identified 5 high-level categories of CT resources: Makerspaces, Coding, Physical Computing/Robotics, Unplugged activities, and Other (which mainly captured media production tools). Of these categories, Coding was the most common with 52 out of 59 participants indicating that they did some sort of coding as part of their library's CT programming. The next most common type of CT programming offered was Physical Computing and Robotics (37) followed by Makerspaces (19). In looking across the types of CT programming offered, we found that most libraries offer more than one type of CT programming—38 of the 59 participants saying their library offers at least 2 types of programs and 13 of those participants offer 3 or more types of programming. Coding and Physical Computing/Robotics were the most likely to be offered together with 32 participants indicating that they had programs focused on both of these topics as part of the programming offered at their library.

In terms of the specific tools and resources being used as part of CT programming in libraries, we found a wide variety across the categories mentioned above. Table 3 presents a summary of the specific resources mentioned by library staff when discussing their CT programming.

One thing that stands out from Table 3 is the large number of libraries using robotics toolkits and the variety of robots. This number is surprising given the relative recency of all these tools and platforms. With the exception of Lego Mindstorms, these tools have

Table 3 Technological resources referenced during participant interviews

Type of resource	Category frequency	Resource (frequency)
Robotics toolkits	49	Sphero (11), Lego Mindstorms (11), Ozobots (10), Code-A-Pillar (5), Dash and Dot (4), Cubettos (3), Bee Bots (3), Finch Robots (2)
Virtual programming tools	21	Scratch (21)
Physical computing devices (circuits)	21	Raspberry Pi (8), Makey Makey (4), Micro:bits (3), Arduino (3), Snap Circuits (3)
Making tools	17	Tinkercad (9), 3D printers (8)
Game-based programming environment	15	Minecraft (7), Bloxels (3), Roblox (3), CodeCombat (2)

Table 4 External resources and organizations referenced during participant interviews

Type of resource	Category frequency	Resource (frequency)
Curricula	45	CS First (13), Girls Who Code (13), Prenda (9), Code.org (4), NCLab (2), CS Unplugged (2), Khan Academy (2)
Initiatives & Support Organizations	34	ALA's Ready to Code (15), Hour of Code (8), YALSA's Train the Trainer and Future Ready (9), National Center for Women & Information Technology (NCWIT) (2)

largely emerged in the last 10 years but have made significant in-roads into libraries. One interesting aspect of these robots and robotics toolkits is a consideration of their versatility and the various ways they can be used as part of library CT programming. "What I have found over the years is that I have to really look at the resource and see if I'm looking at one-off programs, if there are multiple kinds of projects that I can use them for and if there are multiple ways that the kids can create with them because that's not always the case…I tend to like the tools that are a little more open-ended and that offer some tips for ideas" (Caroline, Rural West). The diversity of robotics platforms is in stark contrast to virtual programming tools, where Scratch was the only tool mentioned.

Beyond specific tools, library staff also cited several other forms of resources they rely upon to bring CT programming into their libraries. Initiatives like *Ready to Code* and *Hour of Code* provide grants and planning materials to help with CT programming. Resources like CS First, Girls Who Code, and Prenda provide a curriculum and/or a platform for libraries to use in their programs. The full range of curricular resources and external support organizations, and the frequency that each was cited, are presented in Table 4.

As Table 4 shows, libraries have a high level of dependency on external organizations and initiatives for support in terms of planning, developing, and implementing CT programming. This reflects the newness of the concept of CT and that library staff need guidance and mentorship in providing CT resources and programming to their young patrons. This newness and reliance on external support is also a source of challenges for library staff, an idea we will return to later in the discussion section. For Stephanie (Urban, West) ready-to-use and freely available resources made it easy to start a coding program at her library: "I appreciated that CS First provided all of the materials and all of the tools. I mean, they were totally accessible online, but they also sent me a packet, so I didn't have to think about how to manage the class logistically. They sent the package, and the package has all the little passports for the kids, and it has stickers I can put on at the end of every session to say, 'Congratulations, you learned this new skill,' and all of that was provided. For a library, especially a slightly resource-poor library, the fact that Google was giving us that gift was pretty compelling."

4.2.2 Who is Facilitating Computational Thinking Programming in Libraries?

One way to understand who is designing and leading the CT programming being offered in libraries is to look at who agreed to be interviewed for this study, which was specifically targeting library staff offering CT. The titles of the participants we talked to varied greatly, though many indicated that they worked specifically with youth or with technology and makerspaces. Twenty-four participants had a title/position directly linked to youth, children, or young adults; another 24 had a more general title, like Librarian or Library Associate, Branch Manager, or Director. Nine participants had titles directly linked to technology, makerspaces, or digital learning, like Digital Librarian, Technology Assistant, and Makerspace Specialist.

In terms of prior experience with CT, 11 of our participants indicated that they had previous experience in CT and other STEM concepts through their coursework or previous careers. For example, Matt (Rural, Northeast), explains his background in engineering: "so [CT has] probably been on my radar for quite a long time. Problem-solving in the context of step-by-step analysis is something that's been part of my vocabulary before I studied engineering." One participant, Alyssa (Suburban, West), incorporated the skills she learned in a non-STEM field: "my minor was in project management and so it isn't quite the same terminology of abstraction and all of that, but it is a lot of… 'Here's a thing, we're breaking it down into parts to solve it.'".

Aside from these 11 participants, many staff (29 participants) learned of CT through initiatives like *Libraries Ready to Code* and *Future Ready*. Caroline (Rural, West) used *Ready to Code* as a starting point for looking at different ways to incorporate CT into youth activities in her library: "I was diving deeper into my Libraries Ready to Code Project talking about CT more, in particular, looking at CT with young children, and then I started looking at what CT was and how to support it with young, young children." Twenty-three participants indicated that they mainly taught themselves the skills necessary to host CT programs. When asked where she received training on CT, Molly (Suburban, Southeast) responds, "the school of hard knocks…when everybody's thinking that librarians are just sitting around reading all day, we're not. We're reading and trying to teach ourselves."

Other participants learned about CT and CT concepts by attending various workshops, online webinars, and conference sessions. One participant, Nicole (Suburban, West), talked about how she learned about CT and how she was motivated to expand CT programming at her library: "I was first introduced to the term computational thinking at the YALSA Symposium in November of last year. I attended a program with Google and Google for Libraries I think, and then there were a couple of libraries there that had been participating…talking about outreach that they'd been doing for computational thinking. There was a grant involved. I was like, 'Hey, I was already planning on using my budget money for Week of Code. So, if I can get this grant, Week of Code is going to be that much better.'".

Another strategy used by library staff to better familiarize themselves with CT was to bring in people in their communities that had a STEM background and could teach CT concepts in the library. For example, Brad (Suburban, Northeast) and Molly (Suburban, Southeast) worked with their local high school computer science teachers, Samantha (Rural, West) worked with computer science majors at the local university, Lee (Suburban, Southeast) worked with engineering majors at the local university, Phyllis (Urban, Southeast) worked to bring in students from the local university that were minorities or women in STEM, and Natasha (Rural, Midwest) partnered with the local high school robotics team to have them mentor younger students.

Table 5 The frequency of CT programming in libraries, organized by program type

Program type	Program structure (frequency)
Coding	Repeating sessions (37) Standalone sessions (15)
Physical computing/robotics	Repeating sessions (21) Standalone sessions (17) Standalone & repeating (6)
Makerspace	Repeating sessions (14) Standalone sessions (10) Standalone & repeating (5)

4.2.3 When and How Frequently are Computational Thinking Programs Being Offered?

The final dimension of how we are exploring the nature of CT programming in libraries is to be looking at structural aspects of CT programming, specifically, how and when CT programs are being offered. Our analysis found that CT programs were either standalone (e.g., a single program focused on a single topic or project) or recurring. Twenty-four participants discussed standalone programming with an additional 7 participants saying they held both repeating and standalone programming. Many library staff we interviewed (40 total) discussed repeating CT programming they had organized, like Code Clubs or Maker Mondays. These programs were held weekly (21), monthly (6), or bi-weekly (3). Table 5 provides a summary of the frequency of different types of CT programming in libraries, organized by program type.

4.3 What are the Goals of the Computational Thinking Programming Being Offered by Libraries?

In reviewing the state of CT in libraries, it is important to understand the motivations of the library staff who are leading this effort. In doing so, we gain a picture of both the current state of CT as well as a sense of where library staff are hoping to take their programming and the impact they hope it will have on their young patrons. As part of the interview protocol, participants were asked why they thought learning CT was important for youth and the goals they had for the CT programs they are developing and running. Looking across responses, library staff commonly saw clear alignment between the goals of libraries broadly and the concept of CT. Such views can be seen in responses like: "I look at CT as really fundamentally part of our library's bigger role in supporting literacy and lifelong learning. So, if what I'm doing is to support kids and teens to be literate in a connected world, CT is ultimately a part of that" (Caroline, Rural West) and "I really do believe that [computational thinking] is the next trend with public libraries. They started out with makerspaces and making stuff but I really see this as a foundation for a public library to open up forums where people come in and do this collaborative stuff" (Michelle, Rural Southeast).

To situate the motivations voiced by library staff, we draw from the framework developed by Santo et al., (2019) and Vogel et al., (2017) as part of their work investigating why school district leaders bring computing into their schools. Vogel et al. (2017) identified 7 non-orthogonal motivations: (1) economic and workforce development impacts,

(2) equity and social justice impacts, (3) competencies and literacies impacts, (4) citizenship and civic life impacts, (5) scientific, technological and social innovation impacts, (6) school improvement and reform impacts, and (7) fun, fulfillment and personal agency. In analyzing the motivations we heard from library staff, we see some motivations with close alignment to Vogel et al., some that are similar but with a distinct library flavor, and other motivations entirely distinct from those voiced by school decision-makers.

Economic and workforce development was cited by both district leaders (Santo et al., 2019) and by library staff in similar ways. For example, Lisa (Midwest, Rural) justified her CT programming by saying "For us, we see it as important because most of the jobs that are coming down the pike have some element of CT in them and these kids need to be exposed to that kind of thinking…[CT] is going to apply to so many areas of their life. Even if they don't work in the computing field". This same motivation can be heard in Brad's (Suburban, Northeast) response of "there are so many jobs that go along with computational thinking and it'd be great to get the kids started at a young age and thinking that way so that those doors are open for them later in life". A second overlapping motivation between school and library decision-makers is in the development of competencies and computational literacies. This can be seen in the response "I'm wanting them to develop basic coding skills." (Brittany, Rural Southwest).

Continuing to compare library staff responses to the Vogel et al. (2017) findings, we see additional alignment in motivations but in categories where the role of the library and its distinctness as a cultural and community institution are emphasized. For example, equity through access was also an oft-cited motivation for bringing CT into libraries. Jack (Suburban, Southeast) articulated this motivation clearly in saying "[CT programming] is a good way to introduce kids to things that they might not otherwise be accustomed to, they might not usually see. So, some kids don't even have a computer at home or have Wi-Fi. So, when they come into the library and get to use these tools it kind of really opens them up to other areas of their world that they didn't know they had access to." Another example of this shared alignment with schools but manifest in a slightly different way in libraries can be seen in the response "I would say the goals right now are to introduce CT concepts to kids in really stress-free ways or low-stress ways. They're not being tested. So that and exposing them to different kinds of projects and tools that they might be interested in learning more about" (Caroline, Rural West). Here, we see goals aligned to conceptual learning but emphasizing the informal, low-stakes learning that can happen in libraries. A third example of a library-flavored motivation that aligns with the Vogel et al. framework can be seen in the response: "Just like we have story time so that they can help them learn to read because it's a life skill they're going to need their entire life. The same thing. I consider computational thinking in the same vein. It's a life skill that they're going to need to succeed" (Jane, Rural Southeast). Here we see a motivation for CT grounded in the idea of it as a life skill and motivated by putting it alongside story time and foundational literacy skills.

5 Discussion

5.1 The State of Computational Thinking in Libraries

While equity and access to computing are the most common reasons why libraries are often considered as a place to engage young people in CT (Braun & Visser, 2017;

Taylor et al, 2018), our work offers more motivation for bringing CT into public libraries. We utilized Vogel et al.' (Santo et al., 2019; Vogel et al., 2017) investigation of why school district leaders are bringing computing into their schools as a lens to examine the motivation of bringing CT to libraries. We found some distinct and some similar goals, with the similar goals manifesting differently in the library world. Here, we highlight the distinct goals that we found in our data corpus and explain the connections to the broader learning approaches in public libraries. We found that the connected learning approach that is dominant in public library programming is the framework used in the design and implementation of CT programs in libraries, to enhance academic and economic advancement, as well as civic engagement (Ito et al., 2013). Connected learning programs emphasize interest-driven engagement, which is demonstrated in the expansive portfolio of resources and tools that are currently used in CT programming in libraries, and peer and adult-supported, which is demonstrated by the staff's ability to bring external partners and experts to implement CT programs with them. We found CT programs in libraries still emphasize academic achievement and economic advancement (which are often cited as primary goals in school-based CT learning), but also expand the focus to solving problems that the communities face, facilitating civic engagement among youth as well as combining CT learning with the learning of digital and foundational literacies. This can be deduced from how the library staff conceptualized CT and how they framed the CT activities to solve societal issues (i.e., homelessness, etc.). The amalgamation of the development of CT skills with digital and foundational literacy skills through more commonly known library programs like story time demonstrate how library staff believe that such a two-fold approach (combining CT with digital or foundational literacy) is in alignment with their general goals to support their communities in enhancing learning (ALA, 2008).

The less-structured nature of libraries also affords some advantages that typically do not manifest in school settings. In terms of tools, resources, and curricula used (shown in Tables 3 and 4), we found that library staff have more flexibility to utilize current tools, add a new tool or resource into an existing CT program, or use a preferred curriculum, and can also make on-the-fly changes to their program activities to accommodate these new tools and curricula—almost all done as interest piques among participating youth or as the resources, tools, and curricula become available in libraries. Additionally, working with partners allows libraries to venture into tools, resources, and curricula that are new that comes with more support and guidance to the facilitators who are running these programs (such as the *Ready to Code* initiative) and professional authenticity to computing (such as practices and competencies needed in computing careers and workplaces) (National Academies of Sciences, Engineering, and Medicine, 2021). There is no restriction on following a specific county, state, or national level curriculum. This flexibility contributes to the versatility and diversity of program offerings in libraries.

Some CT programs in libraries include families in their programming that allow youth and their families to bring CT learning into their homes, which allows the cultivation of a network of CT learning opportunities between different settings (in this case libraries and home environments) and library staff provide support for youth to navigate between them (National Academies of Sciences, Engineering, and Medicine, 2021). Recent initiatives indicate that libraries are also bringing programming to where families are such as laundromats, farmers markets, religious centers (Sobel & McClain, 2020)—initiatives such as these can be leveraged to bring CT programs to underserved families that may not be able to get to libraries because of transportation issues.

Another most salient theme that emerged in this study is the organizations that championed CT in libraries through initiatives such as YALSA's Future Ready (ALA, 2016) and ALA's *Ready to Code* (Subramaniam et al., 2019) tend to focus on bringing CT opportunities to rural populations in the U.S. While there have been some initiatives to bring CT to rural schools (Google Inc. & Gallup Inc., 2017) there remain challenges in implementing CT courses such as training teachers and having reliable, modern technological infrastructure (Warner et al., 2019). On the other hand, rural libraries are abundant—The *Public Libraries in the United States Fiscal Year 2016* report, which collected data from more than 98% of public libraries across the country, found that rural areas in the U.S. had more public libraries than cities, suburbs, and towns (IMLS, 2019). Most public libraries (76.52%) served a population area of fewer than 25,000 people (IMLS, 2019). Bringing CT into rural libraries with the unique approaches that we have described above (i.e., connected learning framework, family learning, flexible curriculum), through the support of these organizations and partners increases the participation of rural youth who are often disenfranchised in terms of CT opportunities (Google Inc. & Gallup Inc., 2017).

5.2 Implications for the Design of Computational Thinking Tools and Programs

One recurring theme identified across these analyses is that libraries can be generative contexts for engaging youth in CT. Libraries provide a unique set of opportunities and challenges for those who seek to create learning experiences and tools for these settings. From a designer's perspective, there are a number of implications for the findings presented above. For example, based on the above analysis, CT programming or tools designed for libraries need to balance open-ended exploration and flexibility with structured and scaffolded activities. This can be seen in librarian's expressed desire for tools and resources that provide multiple ways to engage and support a range of activities. At the same time, this work found that many library staff are new to CT and have little formal training with CT. Thus, providing highly scaffolded or pre-defined activities that accompany CT tools or programs is also beneficial for their adoption and use in libraries.

A second implication for those designing tools and technologies for library contexts is creating experiences that meet the needs and goals of libraries. The above analysis shows there are many ways that libraries offer CT programming, including both stand-alone programming (i.e., drop-in CT activities) and recurring weekly or monthly activities. This means CT tools and programs should be able to engage youth who show up once for a drop-in activity with no prior experience while also being able to support continued engagement with youth who return week-after-week to engage in longer, more sustained projects. Likewise, CT tools or programs designed for libraries should consider the myriad of motivations and goals for CT programs in libraries, including developing skills and conceptual knowledge that could lead to future jobs or learning opportunities, supporting the equity and social justice missions of libraries, the goal of preparing a computationally and digitally literate citizenry, and finally, providing positive CT experiences for youth to build their interests and confidence in computational endeavors. While designing a tool, technology, platform, or program that meets all of these goals is certainly challenging, understanding the needs, desires, and constraints of libraries can help in the creation of successful and widely adopted innovations.

5.3 Implications for Library Staff

One of the stated goals for this work is to help support library staff in bringing effective and engaging CT programming into their libraries. The findings from this work can help library staff understand what tools and programs are currently available, as well as provide a picture of how peers are bringing CT into their buildings. While this work does not make specific recommendations on how to bring CT into libraries (nor could it given the diversity of contexts and constraints presented above), this research can help staff understand where they and their institutions fall in the current landscape of CT in libraries, which in turn can help support them in making informed decisions about how to proceed with the goal of offering effective CT programming. For example, the analysis of how library staff conceptualize CT, alongside the presentation of goals of CT programming can help structure library staff reflections on what their goals are and what they want to prioritize when designing their CT programming. Having identified the goals for the CT programming, the design of what tools, technologies, or platforms to use, as well as the CT programming format, can all follow on from the identified desired outcomes. This analysis can then help provide guidance in the form of potential tools to consider based on what is currently widely adopted in the field. In this way, the findings can serve as a guide into the larger world of CT tools, technology, and resources. Collectively, this work can help libraries situate their own programs relative to their peers, both in terms of motivations as well as the practical implementation of the programs they offer.

5.4 Limitations and Future Work

A primary limitation of this research is the sampling method and set of library staff we were able to recruit. Given our sampling method, respondents to our recruitment efforts were likely to be more familiar with CT than a randomly selected library staff member and self-select into learning opportunities about CT. This means our results may present a slightly inaccurate picture of the overall familiarity of library staff with CT. However, we do not feel this undermines the findings as it still reflects how library staff conceptualize CT and what CT programming in libraries look like, but we acknowledge there is more work to be done to better understand how prevalent these CT perceptions and CT programs are across the larger swath of libraries. Second, this research only included libraries in the United States. We acknowledge this is a shortcoming and see an expansion of this work to include libraries from around the world as a next step for this work. A final limitation stems from the rate at which CT tools and programming are shifting. This work captures a snapshot of the current state of CT in Libraries at the time the data was collected but as new technologies, platforms, and initiatives are introduced, we expect the landscape to continue to shift and grow. Again, this does not undermine the contribution of this work, but instead, acknowledges that efforts such as these must be on-going to stay abreast of the changing nature of CT in libraries.

6 Conclusion

Given the growing presence of technology in our world, it is increasingly important that youth develop the knowledge and skills to meaningfully participate in an increasingly digital landscape. These concepts and skills, captured under the term computational thinking, constitute an essential 21st-century skill for all youth to develop. While schools play an important role in this process, libraries have an essential role to play in helping prepare youth for their digital futures. The last decade has seen a rise in efforts to bring CT into libraries, including coding clubs, makerspaces, and several large, coordinated efforts to make libraries hubs for CT learning and exploration. With this work, we take stock of the current landscape of CT in libraries in the United States, documenting how library staff conceptualize CT, reporting the strategies and tools used to bring CT programming into libraries, and identifying the goals of this programming. In doing so, we contribute to the current literature by documenting the ways CT is being integrated into library programming and how library staff think about CT in their libraries. This work helps us understand the current state of CT in libraries, and identifies ways to further support libraries and library staff in helping serve the youth in their communities when it comes to preparing them for their computational futures.

Appendix: Computational Thinking in Libraries Focus Group and Interview Protocol

Introduction

Thank you for participating in our research study today. This interview will last 45—60 min [OR This focus group will last 75–90 min] and we will ask you to answer a few questions about your library, your responsibilities at your library, the population that you serve, current efforts in programming, specifically technology-based programming, existing challenges and constraints, your personal experiences with computational thinking and coding programs, and best practices that your library has for assessing the impact of your programs. You will be providing valuable feedback that will help our research team develop a suite of assessment instruments we hope will capture computational thinking learning that is happening in libraries nationwide. Before we begin, there are a few things we need to go over with you.

Consenting Process

[XXXX]

Start of Interview

1. Could you describe where you work, what your position is, and what you do at your library?
2. Could you describe the size of your library? How many staff work with youth programming?

3. What types of youth programming does your library offer?
4. Could you describe the youth population/s you work with at the library [prompt with gender, race/ethnicity, specific geographical location, SES, academic background, etc.]?

   4.1 Is there any population/s that you would like to serve better?

   Shifting to questions about CT—starting with presenting a definition to provide a shared understanding for future questions.

5. Are you familiar with the term computational thinking? If so, how do you define it? Where have you come across the term? Have you received any training related to computational thinking? If so, where?

   5.1 We are aware that there are so many definitions of CT out there, and it is really hard to find a definition that works for everyone, but the way we will be using the term computational thinking is referring to the concepts and practices associated with using computers and technology to solve problems. CT practices include problem decomposition, developing and using abstractions, debugging, defining algorithms, and concepts related to programming such as loops and variables. Does this definition match how you think about CT?

6. Can you talk about why CT programs are important for children and/or teens?
7. Are you currently running any programs or activities that are related to computational thinking? If so, could you describe them? [Prompt: If they are having trouble realizing that their programs are CT programs, reference back to the answers to question 2 by highlighting the technical programs that they had mentioned]
8. Do your programs build off of each other or are they standalone programs? [probe for how this affects assessment, retention, etc.]
9. What are the youth populations that participate in CT programs that you offer (if you are currently offering CT programs, that is?) [Probe for age, race/ethnicity, gender]
10. What types of resources do you use when you are designing these types of programs? What curriculum or resources do you use? What do you use it for? [Probe for what do you use it for: Is this for your own knowledge development, activity planning, handouts, or things you have the kids use/do? Technology used?]. Why do you use this resource?
11. What challenges or constraints do you face with your CT programming in your library?
12. If you have run CT programming—what are your goals for those programs? If you haven't run CT programming—if you did, what would your goals be?
13. What skills and attitudes are you hoping to develop as a result of the program [Probe: dig more into what they are interested in seeing evidence of in their programs (shifts or development in knowledge, perceived futures, perceptions, skills, etc.]?
14. Next, we are going to ask you a few questions about how you know if your programs have been successful.
15. How do you know if your program is successful? [Probes: What does success mean to you? How do you measure learning? What do you use to measure the success of your programs? How did you decide on this set of measures? Were there other measures you considered?]

16. What do you do with the information/data you collect? Who is the audience [prompt for administrators, funders, internal professional learning and program improvement]? How do you share the success of your programs with your stakeholders?
17. How else would you like to measure the success of your program? Why are these metrics useful?

Conclusion

"Thank you very much for participating in the session today. Your feedback will be very beneficial and help our team understand how to improve youth-connected learning experiences in libraries. Unless you have any other thoughts, I will now turn off the recording."

Funding This work was supported by the Institute of Museum and Library Services, USA under Grant # LG-14-19-0079-19.

Data availability This manuscript has no associated data or the data will not be deposited.

Declarations

Conflict of interests The authors have no relevant financial or non-financial interests to disclose.

References

Abelson, H., & diSessa, A. A. (1986). *Turtle geometry: The computer as a medium for exploring mathematics*. The MIT Press.

ALA. (2008). *Mission & Priorities*. American Library Association. http://www.ala.org/aboutala/missionpriorities.

ALA. (2016). *Future ready with the library*. American Library Association. http://www.ala.org/yalsa/future-ready-library.

ALA. (2020). *State of America's libraries report 2020*. American Library Association. http://www.ala.org/news/state-americas-libraries-report-2020.

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: implications for teacher knowledge. *Journal of Educational Technology & Society, 19*(3), 47–57.

Barron, B. (2006). Interest and self-sustained learning as catalysts of development: A learning ecology perspective. *Human Development, 49*(4), 193–224.

Berland, M., & Lee, V. R. (2011). Collaborative strategic board games as a site for distributed computational thinking. *International Journal of Game-Based Learning, 1*(2), 65–81. https://doi.org/10.4018/ijgbl.2011040105

Bertot, J. C., Sarin, L. C., & Percell, J. (2015). Re-envisioning the MLS: Findings, issues, and considerations. *University of Maryland, College Park, College of Information Studies. Accessed August*, *27*, 2015.

Boyatzis, R. E. (1998). *Transforming qualitative information: Thematic analysis and code development*. Sage.

Braun, L., Ciotti, S., & Peterson, S. (2017). *Make do share: Sustainable STEM programming for and with youth in public libraries*. Kitsap Regional Library. http://www.krl.org/makedoshare.

Braun, L., & Visser, M. (2017). Ready to code: Connecting youth to CS opportunity through libraries. *Libraries Ready to Code*.

Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*. American Education Researcher Association, Vancouver, Canada. http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf.
Curzon, P., Bell, T., Waite, J., & Dorling, M. (2019). Computational thinking. In *The Cambridge handbook of computing education research* (pp. 513–546).
Davis, K., Subramaniam, M., Hoffman, K. M., & Romeijn-Stout, E. L. (2018). Technology use in rural and urban public libraries: Implications for connected learning in youth programming. In *Proceedings of the connected learning summit (CLS '18)* (pp. 47–56).
Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM, 60*(6), 33–39.
Dresang, E. T. (2013). Digital age libraries and youth: Learning labs, literacy leaders, radical resources. In *The information behavior of a new generation: Children and teens in the 21st century* (pp. 93–116).
Ehsan, H., Rehmat, A. P., & Cardella, M. E. (2020). Computational thinking embedded in engineering design: Capturing computational thinking of children in an informal engineering design activity. *International Journal of Technology and Design Education*. https://doi.org/10.1007/s10798-020-09562-5
Einarsson, Á. M., & Hertzum, M. (2020). How is learning scaffolded in library makerspaces? *International Journal of Child-Computer Interaction, 26*, 100199. https://doi.org/10.1016/j.ijcci.2020.100199
Garmer, A. K. (2014). *Rising to the challenge: Re-envisioning public libraries*. Aspen Institute.
Garvin, M., Killen, H., Plane, J., & Weintrop, D. (2019). Primary school teachers' conceptions of computational thinking. In *Proceedings of the 50th ACM technical symposium on computer science education* (pp. 899–905). https://doi.org/10.1145/3287324.3287376.
Google Inc., & Gallup Inc. (2017). *Computer science learning: Closing the Gap: Rural and Small Town School Districts*. Google & Gallup. http://goo.gl/hYxqCr.
Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher, 42*(1), 38–43.
Guidara, M. (2018). Rethinking computational thinking for public libraries' youth programs: Challenges and recommendations. *Pennsylvania Libraries: Research & Practice, 6*(2), 75–85.
Hadad, R., Thomas, K., Kachovska, M., & Yin, Y. (2020). Practicing formative assessment for computational thinking in making environments. *Journal of Science Education and Technology, 29*(1), 162–173. https://doi.org/10.1007/s10956-019-09796-6
Hamilton, M. M., Clarke-Midura, J., Shumway, J. F., & Lee, V. R. (2018). *An initial examination of designed features to support computational thinking in commercial early childhood toys* (Vol. 2).
Harel, I., & Papert, S. (1990). Software design as a learning environment. *Interactive Learning Environments, 1*(1), 1–32.
Hoffman, K., Subramaniam, M., Kawas, S., Scaff, L., & Davis, K. (2016). *Connected libraries: Surveying the current landscape and charting a path to the future*. The Connected Lib Project. https://connectedlib.ischool.uw.edu/.
Holbert, N., & Wilensky, U. (2011). FormulaT racing: Designing a game for kinematic exploration and computational thinking. In *Proceedings of the 7th games, learning, & society conference*.
Horn, M. S., Brady, C., Hjorth, A., Wagh, A., & Wilensky, U. (2014a). Frog pond: A codefirst learning environment on evolution and natural selection. In *Proceedings of the 2014a conference on interaction design and children* (pp. 357–360). http://dl.acm.org/citation.cfm?id=2610491.
Horn, M. S., Solovey, E. T., Crouser, R. J., & Jacob, R. J. K. (2009). Comparing the use of tangible and graphical programming languages for informal science education. In *Proceedings of the 27th international conference on human factors in computing systems* (pp. 975–984).
Horn, M. S., Weintrop, D., & Routman, E. (2014b). Programming in the pond: A tabletop computer programming exhibit. In *Proceedings of the extended abstracts of the 32nd annual ACM conference on human factors in computing systems* (pp. 1417–1422). https://doi.org/10.1145/2559206.2581237.
IMLS. (2019). *Public libraries in the United States—Fiscal Year 2016*. Institute of Museum and Library Services.
Ito, M., Arum, R., Conley, D., Gutiérrez, K., Kirshner, B., Livingstone, S., Michalchik, V., Penuel, W. R., Peppler, K., Pinkard, N., Rhodes, J., Salen Tekinbas, K., Schor, J., Sefton-Green, J., & Watkins, S. C. (2020). *The connected learning research network: Reflections on a decade of engaged scholarship*. Connected Learning Alliance. https://clalliance.org/publications/the-connected-learning-research-network-reflections-on-a-decade-of-engaged-scholarship/.
Ito, M., Gutiérrez, K., Livingstone, S., Penuel, B., Rhodes, J., Salen, K., Schor, J., Sefton-Green, J., & Watkins, S. C. (2013). *Connected learning: An agenda for research and design*. Digital Media and Learning Research Hub. http://dmlhub.net/.
Kay, A., & Goldberg, A. (1977). Personal dynamic media. *Computer, 10*(3), 31–41.

Koh, K., & Abbas, J. (2015). Competencies for information professionals in learning labs and makerspaces. *Journal of Education for Library and Information Science Online*, *56*(2), 114–129. https://doi.org/10.12783/issn.2328-2967/56/2/3.

Koh, K., Abbas, J., & Willett, R. (2018). Makerspaces in libraries. *Reconceptualizing Libraries: Perspectives from the Information and Learning Sciences*, 17–36.

Lee, T. Y., Mauriello, M. L., Ahn, J., & Bederson, B. B. (2014). CTArcade: Computational thinking with games in school age children. *International Journal of Child-Computer Interaction*. https://doi.org/10.1016/j.ijcci.2014.06.003

Lee, V. R. (2019). Libraries will be essential to the smart and connected communities of the future. In V. R. Lee & A. L. Phillips (Eds.), *Reconceptualizing libraries: Perspectives from the information and learning sciences*. Routledge.

Lee, V. R., & Phillips, A. L. (2018). *Reconceptualizing libraries: Perspectives from the information and learning sciences*. Routledge.

Lee, V. R., & Recker, M. (2018). Paper circuits: A tangible, low threshold, low cost entry to computational thinking. *TechTrends, 62*(2), 197–203.

Martin, C. (2017). Libraries as facilitators of coding for all. *Knowledge Quest, 45*(3), 46–53.

Mesiti, L. A., Parkes, A., Paneto, S. C., & Cahill, C. (2019). Building capacity for computational thinking in youth through informal education. *Journal of Museum Education, 44*(1), 108–121. https://doi.org/10.1080/10598650.2018.1558656

Metcalf, L. E., & Anderson, J. L. (2020). *Hidden no more: Using museum exhibits of underrepresented scientists to engage students in computational thinking* (pp. 1416–1421). https://www.learntechlib.org/primary/p/215907/.

Morgan, D. L. (1996). Focus groups. *Annual Review of Sociology, 22*(1), 129–152.

National Academies of Sciences, Engineering, and Medicine. (2021). *Cultivating interest and competencies in computing: authentic experiences and design factors* (p. 25912). National Academies Press. https://doi.org/10.17226/25912.

National Research Council. (2010). *Report of a workshop on the scope and nature of computational thinking*. The National Academies Press.

National Research Council. (2011). *Report of a workshop of pedagogical aspects of computational thinking*. The National Academies Press.

Papert, S. (1972). Teaching children to be mathematicians versus teaching about mathematics. *International Journal of Mathematical Education in Science and Technology, 3*(3), 249–262.

Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. Basic books.

Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer*. Basic Books.

Peppler, K., Halverson, E., & Kafai, Y. B. (2016). *Makeology: Makerspaces as learning environments* (Vol. 1). Routledge.

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

Pinkard, N., Martin, C. K., & Erete, S. (2020). Equitable approaches: Opportunities for computational thinking with emphasis on creative production and connections to community. *Interactive Learning Environments, 28*(3), 347–361. https://doi.org/10.1080/10494820.2019.1636070

Prato, S. C. (2017). Beyond the computer age: A best practices intro for implementing library coding programs. *Children and Libraries, 15*(1), 19–21.

Real, B., & Rose, R. N. (2017). Rural libraries in the United States: Recent strides, future possibilities, and meeting community needs. Office for Technology Policy at the American Library Association. Accessed July, 17, 2020.

Resnick, M., Silverman, B., Kafai, Y., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., & Silver, J. (2009). Scratch: Programming for all. *Communications of the ACM, 52*(11), 60.

Santo, R., Vogel, S., & Ching, D. (2019). *CS for What? Diverse visions for computer science education in practice*. CSforALL.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review, 22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Smagorinsky, P. (2008). The method section as conceptual epicenter in constructing social science research reports. *Written Communication, 25*(3), 389–411.

Sobel, K., & McClain, A. (2020). *Libraries: Connecting Family Learning Across Settings* (p. 2). Joan Ganz Cooney Center. https://joanganzcooneycenter.org/wp-content/uploads/2020/03/jgcc_librariesconnectinglearning.pdf.

Subramaniam, M., Hoffman, K.M., Davis, K. & Pitt, C. (2021). Designing a connected learning toolkit for public library staff serving youth through the design-based implementation research method. *Library and Information Science Research, 43*(1).

Subramaniam, M., Kodama, C., Baylen, D., Burton, M., Fabicon, J. K., Hincks, K., Moniz, R., Smith, D., & Visser, M. (2019). *Computational thinking in libraries: Case studies of youth programs in action.* The American Library Association's Office for Information Technology Policy.

Subramaniam, M., Koren, N., Morehouse, S., & Weintrop, D. (2022). Capturing computational thinking in public libraries: An examination of assessment strategies, audience, and mindset. Journal of Librarianship and Information *Science*, 096100062210841.

Subramaniam, M., Scaff, L., Kawas, S., Hoffman, K. M., & Davis, K. (2018). Using technology to support equity and inclusion in youth library programming: Current practices and future opportunities. *The Library Quarterly, 88*(4), 315–331.

Taylor, N. G., Moore, J., Visser, M., & Drouillard, C. (2018). Incorporating computational thinking into library graduate course goals and objectives. *School Library Research*, *21*.

Thompson, K. M., Jaeger, P. T., Taylor, N. G., Subramaniam, M., & Bertot, J. C. (2014). *Digital literacy and digital inclusion: Information policy and the public library*. Rowman & Littlefield.

Tsarava, K., Moeller, K., & Ninaus, M. (2018). Training computational thinking through board games: The case of crabs & turtles. *International Journal of Serious Games*, *5*(2), 25–44. https://doi.org/10.17083/ijsg.v5i2.248

Tzou, C., Bell, P., Bang, M., Kuver, R., Twito, A., & Braun, A. (2019). Building expansive family STEAM programming through participatory design research. In V. R. Lee & A. L. Phillips (Eds.), *Reconceptualizing libraries: Perspectives from the information and learning sciences*. Routledge.

Vickery, J. (2014). Youths teaching youths: Learning to code as an example of interest-driven learning. *Journal of Adolescent & Adult Literacy, 57*(5), 361–365.

Vogel, S., Santo, R., & Ching, D. (2017). Visions of computer science education: Unpacking arguments for and projected impacts of CS4All initiatives. In *Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education—SIGCSE '17* (pp. 609–614). https://doi.org/10.1145/3017680.3017755.

Warner, J. R., Fletcher, C. L., Torbey, R., & Garbrecht, L. S. (2019). Increasing capacity for computer science education in rural areas through a large-scale collective impact model. In *Proceedings of the 50th ACM technical symposium on computer science education* (pp. 1157–1163). https://doi.org/10.1145/3287324.3287418.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*(1), 127–147. https://doi.org/10.1007/s10956-015-9581-5

Weintrop, D., Holbert, N., Horn, M. S., & Wilensky, U. (2016). Computational thinking in constructionist video games. *International Journal of Game-Based Learning, 6*(1), 1–17. https://doi.org/10.4018/IJGBL.2016010101

Weintrop, D., Morehouse, S., & Subramaniam, M. (2021). Assessing computational thinking in libraries. *Computer Science Education*, 1–22. https://doi.org/10.1080/08993408.2021.1874229.

Wilensky, U. (1999). *NetLogo*. Center for Connected Learning and Computer-Based Modeling, Northwestern University. http://ccl.northwestern.edu/netlogo.

Wing, D., & Meyers, E. (2014). Easy as Pi: Designing a library program to support computational thinking in preteens. *BCLA Browser: Linking the Library Landscape*, *6*(3).

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35.

Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM, 60*(4), 55–62. https://doi.org/10.1145/2994591

Yu, J., & Roque, R. (2019). A review of computational toys and kits for young children. *International Journal of Child-Computer Interaction, 21*, 17–36. https://doi.org/10.1016/j.ijcci.2019.04.001