Integration of computational thinking into English language arts

Authors
Jacob, Sharin R
Parker, Miranda C
Warschauer, Mark

Publication Date
2022

DOI
https://doi.org/10.1145/3507951.3519288

Integration of Computational Thinking Into English Language Arts

Sharin Rawhiya Jacob, Miranda C. Parker, and Mark Warschauer, *University of California, Irvine*

Corresponding Author: Sharin Rawhiya Jacob, *Sharinj@uci.edu*

Abstract

This paper describes the development and implementation of a yearlong integrated English Language Arts (ELA) and computational thinking (CT) curriculum that has been adapted to meet the needs of multilingual students. The integration of computational thinking into K-12 literacy instruction has only been examined in a handful of studies, and little is known about how such integration supports the development of CT for multilingual students. We conducted a qualitative case study on curricular implementation in a general education classroom with large numbers of students designated as English learners. Results from detailed field notes revealed that the strategic application of instructional practices was implemented in the service of building on students' existing literacy skills to teach CT concepts and dispositions. The CT and literacy framework put forth in this study can be used as an analytic framework to highlight how instructional strategies mobilize the existing literacy and CT resources of linguistically diverse students. Based on our findings, we discuss recommendations for future integrated ELA-CT curricula.

While the integration of computational thinking (CT) into science, technology, engineering, and mathematics (STEM) education has been well studied (Jona et al., 2014; Sengupta et al., 2018; Weintrop et al., 2016), there is a smaller but growing body of work on CT and literacy integration (Jacob et al., 2018; Burke & Kafai, 2012; Kafai et al., 2020; Vogel et al., 2020). There are several affordances to engaging diverse learners when combining CT and literacy instruction. Programming in narrative genres may foster literacy development and technological fluency while motivating students who may not otherwise identify with computer science (CS; Burke & Kafai, 2012). This can facilitate the kinds of inquiry, cultural and community engagement, and social recognition that are integral to fostering identity development in STEM (National Research Council [NRC], 2014).

Computational thinking and literacy integration is particularly beneficial in elementary grades, as instructional minutes allotted to STEM are extremely limited, especially for students who are second language learners (Dorph et al., 2011). While the value of focusing on language and literacy instruction in early grades is undisputed, integration of CT within the language arts curriculum can provide a way to overcome STEM instructional time constraints, allowing students to get vital early exposure to CS while also supporting their language development.

This paper describes the implementation of an English Language Arts (ELA)-focused curriculum to support learning and positive identification with CS among multilingual elementary school students. We first describe the model of computational literacies we draw on and then describe the curriculum that forms the basis of the intervention and study.

We address the following research question:

A. What strategies are used by upper elementary teachers to integrate CT into literacy and language instruction?
B. How does applying the CT and literacy framework advance our understanding of how to leverage multilingual students' literacy resources to develop their computational thinking skills?

Computational Literacies

Our study draws from Jacob and Warschauer's (2018) model of computational literacy, which situates computational thinking as a fundamental literacy required for full societal participation (cf. diSessa, 2000; Wing, 2006). This model proposes three dimensions for 1) characterizing the relationship between computational thinking and literacy (i.e., computational thinking *as* literacy), 2) examining how students' existing literacy skills can be leveraged to foster computational thinking (i.e., computational thinking *through* literacy), and 3) discussing the ways in which computational thinking skills foster literacy development (i.e., literacy *through* computational thinking; Jacob & Warschauer, 2018; see Figure 1).

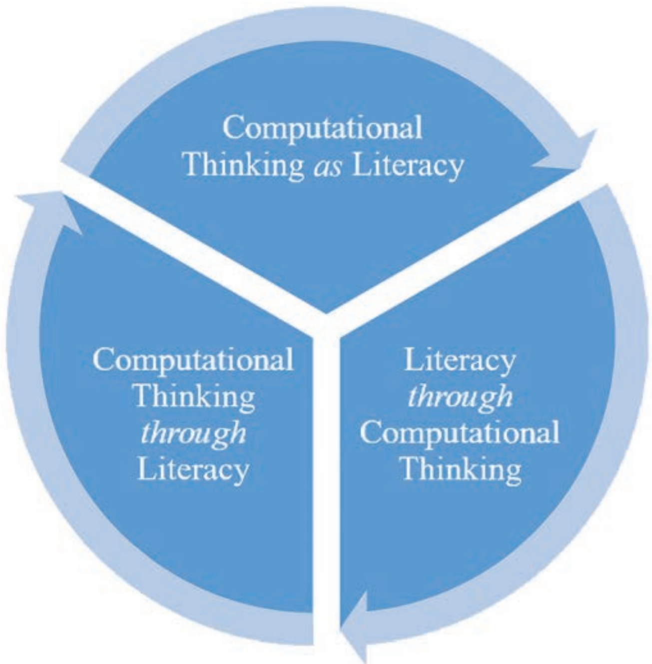

Figure 1. A Three-Dimensional Framework for Understanding Computational Thinking and Literacy (Jacob & Warschauer, 2018)

For the purpose of this paper, we focus on the second component of the computational thinking and literacy framework: computational thinking *through* literacy. To this end, we examine how students leverage their existing literacy skills as a mechanism for learning computational thinking. Integrating computational thinking into ELA content has multiple affordances for CT learning. Evidence suggests that learning to read and write and to code can go hand in hand (Peppler & Warschauer, 2011; Bers, 2019). The several interlocking features of coding and literacy draw children's attention to symbol-meaning relationships. For example, students interact with text in multiple ways as they use Scratch and leverage their knowledge of multimodal signifiers to assemble programs. These relationships offer a highly engaging and supportive environment for children with emerging literacies to demonstrate their skills and abilities (Peppler & Warschauer, 2011).

Additionally, informational and narrative genres capture the semiotic process related to computing. To illustrate, Burke and Kafai (2012) leveraged students' knowledge of the writing process (i.e., drafting, revising, editing) to engage them in designing computational artifacts (i.e., (design, troubleshooting, debugging). Similarly, De Souza et al. (2011) compared students' narrative accounts of programming games to their design process, paying specific attention to verbal structures. Findings indicated that at first students used transitive verb based narrative accounts to design games, and over time they began to use intransitive verbal structures that more closely resembled programming languages For example, A typical student characterization of a game "the hunter killed the monkey" was actually programmed as "the monkey disappears when it touches the hunter." Results such as these suggest that students' existing literacy skills can be mobilized to develop their computational thinking skills.

The CS-ELA Integrated CT Curriculum

Elementary schools with large percentages of multilingual students, not surprisingly, devote large amounts of instructional time to improving students' English skills. This makes it challenging to introduce non-core curriculum, such as CS. Indeed, research has shown that science, let alone CS, is rare in high-ELL schools and districts (Gomez-Zwiep, 2017). Our project has addressed this challenge by adapting the Creative Computing Curriculum (Brennan et al., 2014) for integration into ELA instruction. The curriculum--called Elementary Computing for All--exploits the affordances of Scratch for learning to decode and code stories of the same genres that are emphasized in traditional narrative and informative texts in elementary school. It also integrates age-appropriate readings about diverse pioneers in CS, thus strengthening the connection to reading while also providing culturally relevant support. In this way, STEM identity is developed as children learn about diverse computer scientists and code stories about their own lives and communities.

The storybooks integrated into the curriculum teach not only computational thinking concepts but also key dispositions that foster student success in computing. In 2011, the International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA) outlined specific dispositions or mindsets that are fundamental to student success in computational thinking including 1) confidence in dealing with complexity, 2) persistence in working with difficult problems, 3) tolerance for ambiguity, and 4) the ability to deal with open ended problems (ISTE & CSTA, 2011). The storybooks in our curriculum teach these dispositions in culturally and age appropriate ways. For example, students read *The Most Magnificent Thing*, a storybook about a young girl who, through engaging in making activities, acquires positive dispositions and approaches to computing. The protagonist of the book desires to construct a computational artifact for her dog. Throughout the design process, she abstracts her model, decomposes her problem, implements her solutions, debugs her errors, and engages in iterative problem solving to arrive at a "magnificent" solution. To this end, the storybook teaches both computational thinking concepts such as abstraction, iteration, decomposition, and debugging as well as dispositions that enable students to become successful computational thinkers. The big idea of the story, having a growth mindset, is operationalized through examples of the protagonist dealing with complex problems, persisting through mistakes, and tolerating ambiguity. Storybooks such as these provide affordances for teaching both the computing concepts necessary for learning the discipline as well and dispositions that foster successful computational thinkers.

Linguistic Scaffolding

Researchers and practitioners worked collaboratively to develop additional language scaffolding to amplify the curriculum's effectiveness with multilingual students, following effective practices recommended by a national panel (National Academies of Science, Engineering, and Medicine [NASEM], 2018). First, the revised curriculum integrates CS and ELA tasks to **engage students in disciplinary practices**. Students explore and modify existing programs before creating their own projects. These kinds of structured inquiry-based science approaches provide a powerful mechanism for providing authentic contexts for language use (NRC, 1996) while making instruction more engaging, concrete, and meaningful for multilingual students (Janzen, 2008; NRC, 2012; Rosebery & Warren, 2008). Computer science disciplinary activities and learning goals are aligned with standards to guide teachers (see Table 1 for an example).

Second, the revised curriculum **encourages rich classroom discourse** through explicit suggestions of collaborative activity formats to invite students to use their everyday sense-making and disciplinary language in multiple contexts (Shea & Shanahan, 2011).

Third, strategies that teachers can use to **build on students' existing resources** (i.e., cultural, linguistic, semiotic, embodied) to acquire proficiency in language and CS are explicitly stated in the curriculum and during professional development. For example, the curriculum

Table 1. Sample Learning Goals That Integrate Grade 4 Common Core ELA, English Language Development, and Computer Science Teachers Association Standards

Activity: Students program a story about their lives, families, or communities		
Computer Science Concepts: Loops, Sequences, Conditionals		
Computer Science Teachers Association (CSTA) Standards		
CSTA 1B-AP-10	Create programs that include sequences, events, loops, and conditionals	
CSTA 1B-AP-13	Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences	
CSTA 1B-AP-15	Test and debug a program or algorithm to ensure it runs as intended	
English Language Development (ELD) Standards		
Emerging	**Expanding**	**Bridging**
3. Offering opinions Negotiate with or persuade others in conversations using basic learned phrases (e.g., I think) as well as open responses in order to gain and/or hold the floor.	**3. Offering opinions** Negotiate with or persuade others in conversations using a variety of learned phrases (e.g., That's a good idea. However…) as well as open responses, in order to gain and/or hold the floor.	**3. Offering opinions** Negotiate with or persuade others in conversations using a variety of learned phrases (e.g., That's a good idea. However…) as well as open responses in order to gain and/or hold the floor, elaborate on an idea, and provide different opinions.
11. Supporting opinions Offer opinions and provide good reasons (e.g., My favorite book is X because X) referring to the text or to relevant background knowledge.	**11. Supporting opinions** Offer opinions and provide good reasons and some textual evidence or relevant background knowledge (e.g., paraphrased examples from text or knowledge of content).	**11. Supporting opinions** Offer opinions and provide good reasons with detailed textual evidence or relevant background knowledge (e.g., specific examples from text or knowledge of content).
Corresponding English Language Arts Standards		
CCSS.ELA-L.SL.4.1 **CCSS.ELA-L.SL.4.4** **CCSS.ELA-L.SL.4.6** **CCSS.ELA-L.W.4.9**	**Engage** effectively in a range of collaborative discussions with diverse partners, building on others' ideas and expressing their own clearly. **Report** on a topic or text, tell a story, or recount an experience in an organized manner, using appropriate facts and relevant, descriptive details to support main ideas or themes; speak clearly at an understandable pace. **Differentiate** between contexts that call for formal English (e.g., presenting ideas) and situations where informal discourse is appropriate (e.g., small-group discussion); use formal English when appropriate to task and situation. **Draw** evidence from literary or informational texts to support analysis, reflection, and research.	

and professional development include tips for teacher "talk moves'' (Michaels & O'Connor, 2015), namely asking for clarification and leveraging students' own ways of explaining to guide them towards more formal language and advanced CS concepts.

Fourth, visualizations and physical, unplugged activities are built into the curriculum to **engage students in multiple modalities**, including linguistic modalities of talk and text, as well as nonlinguistic modalities such as gestures, pictures, and symbols, to better teach key academic vocabulary and CT concepts (cf. Lee et al., 2019).

Fifth, the curriculum **provides explicit focus on how language functions in the discipline** by providing language frames to teachers for use by students during peer feedback and pair programming, and while asking for assistance (see example in Table 2).

METHODS

Researchers at Western University (pseudonym) and educators in a large urban school district joined together in a research-practice partnership to iteratively develop and implement the curriculum. The district has among the highest percentages in the nation of Latinx students (93%), low-income learners (89.7% receiving free or reduced-price lunch), and students designated as English language learners (62.7% in the elementary grades). Ordinary elementary school teachers in the district taught the curriculum in their own classes after a one-week professional development program in the summer that taught them about Scratch, computational thinking, equity issues in CS education, and the CT-ELA approach.

Though broader data were collected for the larger study, in this paper we only focus on the instructional strategies carried out by teachers to integrate CT and ELA instruction that meets the needs of multilingual students. Thus, we analyze structured notes from weekly classroom observations. For each field observation, two researchers took detailed field notes on teachers' instructional moves, students' interaction, and computing tasks and activities. Four Ph.D. students and three undergraduates observed teachers' classes when they integrated CT and literacy lessons. All lessons were audio recorded and transcribed.

These data were analyzed through open coding in iterative cycles. Two researchers collaborated to assign initial codes to excerpts of text that pertained to strategies used by teachers to integrate CT and ELA content (Hsieh & Shannon, 2005), paying specific attention to instructional practices that are effective for engaging multilingual students in STEM (NASEM, 2018). After coding 25% of field notes, the researchers met to combine, split, and categorize codes based on initial findings. After this discussion, the first author applied the consolidated codes to the rest of the data, generating new codes when they were pertinent to the research questions. After coding all of the field notes, two researchers (first and second author) randomly selected 10% of the data to conduct an interrater reliability check and achieved 83% agreement. The two researchers then met to discuss differing codes and redefine each of the codes. After revising the codebook, they reapplied the modified codes and reached 94% agreement.

RESULTS

All the teachers in our study were able to successfully teach the curriculum and carry out appropriate strategies for students designated as English learners that integrated CT and ELA in the classroom (see Table 3). To illustrate this, we present a case from one classroom taught by Jenny (pseudonym), which was a general education classroom of predominantly Latinx and low-income students designated as English learners.

Table 2. Computer Science Language Functions

Teacher Activities	Student Discourse			CS Concepts (Language Function)
	Emerging	Expanding	Bridging	
Remind students to think about the events that will cause each action to happen in their project, which programs will run parallel to each other, and how their project will reset once it has finished running.	I need help with __. __ caused __ to happen. __ and__ are running at the same time. I used __ to reset the program.	I am having difficulty with __. __ is the event that caused __ to happen. __ and __ are running parallel to each other. I used __ to initialize the program.	Could you help me fix the following challenge in my code __? The event that caused __ to happen is __. __ and __ are running parallel to each other/ simultaneously/at the same time. __ caused the program to initialize.	Debugging, events, initialization, parallelism (Describing, comparing)

Table 3. Coding Framework Excerpt

Categories	Strategies for Activating Prior Knowledge	Strategies for Asking Questions	Strategies for Providing Direct Instruction	Strategies for Providing Language Support
Sample Codes and Definitions	Leveraging Students' Background Knowledge *Applying students' existing knowledge to lesson content.* Building on Students' Personal Experiences *Connecting lessons to students' personal experiences.*	Using Questions to Foster Higher Order Thinking *Asking higher order questions (i.e., analysis, evaluation) instead of recall of comprehension checks.* Using Questions to Make Interdisciplinary Connections *Asking how one subject is similar to another (e.g., using elements of storytelling to describe coding processes).* Using Questions to Elicit Big Idea *Asking how instructional materials relate to the big idea of the lesson.*	Discussing Computational Concepts *Discussing computational concepts (i.e., abstraction, algorithms) and programming concepts (i.e., sequence, loops, conditionals).* Pre-Teaching Lesson Vocabulary *Introducing lesson vocabulary in multiple modalities at beginning of lesson.*	Facilitating Discourse Through Collaboration *Engaging students in peer-to-peer or teacher-student-student talk to build on students' existing resources.* Prompting Students to Use Sentence Frames *Using sentence frames as prompts to provide language support, guidance, and to encourage elaboration.* Encourage Students to Use CS Language During Reflection *Encouraging students to use CS language gradually on their own.*

Strategies Used for CT-ELA Integration

Jenny's most frequently used strategy included multiple questioning techniques, and she made a point to integrate ELA reading strategies with CT lessons. For example, after reading *The Most Magnificent Thing* to her students, she used questioning techniques to check students' understanding of key computational thinking concepts such as sequencing, decomposition, debugging, and abstraction. She also used questions to elicit big ideas, such as developing a growth mindset. To illustrate, after the protagonist of the story *The Most Magnificent Thing* finished designing her computational artifact, the teacher asked: *"Was it perfect?"* The students responded: "No!" Then the teacher asked, *"But did it do the job?"* and the whole group responded *"Yes!"* In this example, she underscored for her students the idea that while they can always improve their work, they should also be proud of the artifacts that they have created. Research corroborates the idea that the design process is iterative and emphasis should be placed on process over product when developing computational artifacts (Ryoo et al., 2015). Finally, Jenny's use of multiple questioning techniques facilitated comprehension of CT and literacy content by providing opportunities for students to experience ideas in multiple ways. She primarily questioned students during whole group activities and used specific techniques related to ELA instruction such as encouraging higher order thinking (i.e., providing supporting evidence), elaborating components of storytelling (i.e., students identify plot, characters, setting, conflict, resolution), and invoking the big idea (i.e., identifying the main idea of the lesson).

Jenny also facilitated student discourse by engaging them in collaboration during pair programming activities. For example, she instructed her students to provide constructive feedback to their peers, even if their peers' projects contained mistakes. This helped to normalize the making of mistakes in the classroom and foster persistence in the face of challenges. Another strategy Jenny used was the activation of prior knowledge, which involves priming students' existing knowledge and providing prerequisite knowledge for students to understand lesson concepts. To this end, Jenny would reference previous CT lessons to connect to the current lesson she was teaching. This strategy is essential to providing a foundation for multilingual students to assimilate new information (Lee & Fradd, 1998; Turner & Bustillos, 2017). Jenny also promoted the use of discipline-specific discourse by fostering interaction, prompting student reflection during whole group discussion, and modeling the use of CS language during whole group instruction.

Applying a CT and Literacy Framework Through CT-ELA Integration

We present a vignette that explores how the instructional moves employed by Jenny apply the CT and literacy framework (Jacob & Warschauer, 2018) to integrate CT-ELA instruction for multilingual students. The purpose of this section is to advance our understanding of how teacher moves can benefit culturally and linguistically diverse students in a CT-ELA integrated curriculum.

Teaching CT and Literacy in Jenny's Diverse General Education Classroom

In the excerpt below (Table 4), Jenny reads *The Most Magnificent Thing* to her students and pauses the story multiple times to question her students to emphasize the key idea.

In this excerpt, Jenny is teaching *computational thinking through literacy* by leveraging students' knowledge of storytelling and narrative devices to engage them in productive discussion of computing concepts and dispositions. Using well-established techniques such as making predictions and discussing main ideas (Wright & Gotwals, 2018) allows Jenny's students to check her students' understanding of CT and literary concepts, through whole group interaction that is broken down into meaningful chunks. Through her questions, Jenny encourages students to engage in several CT concepts and practices, including sequences ("What happened next?"), abstraction ("What did she notice about all of those things?"), and experimenting and iterating ("Was it perfect?"). With this process she simultaneously teaches literary themes (i.e., plot, character development, conflict, resolution, theme), CT concepts (i.e., iteration, testing, debugging, design process), and positive attitudes and dispositions towards CT (i.e., growth mindset, confidence, perseverance). In her next lesson, Jenny moves on to apply the idea of a growth mindset to students' programming tasks, encouraging students to iterate and debug

Table 4. Audio Transcript of Jenny Teaching The Most Magnificent Thing

Speaker	Audio Transcript
Jenny:	(teacher pauses story) Why is she quitting? Talk to your partner. Why is she quitting? Tell me, why is she giving up? (students are busy discussing with one another)
Student 1:	It is too hard…
Student 2:	Not the way she wants it to be…
Student 3:	Maybe because what she is thinking it is not possible because it is hard (teacher resumes story then teacher pauses story again)
Jenny:	So tell me first of all, what was the problem with what she was building? What was she building?
Student 4:	A robot…
Student 5:	A car…
Jenny:	(Jenny plays the story to find out what she is building) What did she do? What happened first? What did she do first?
Student 4:	She got mad.
Jenny:	What happened next? Did she just stay mad and give up? What happened next?
Student 3:	She took her dog out for a walk and saw all that she did and what she gave up.
Student 6	So she looked at all of her work that she thought was wrong.
Jenny:	And what did she notice about all of those things?
Student 3:	There were pieces that she liked.
Student 7:	There were the right pieces that she made.
Jenny:	So she had to do what? To her thinking? She had to do what to her thinking?
Student 8:	She had to look at her invention.
Student 9:	Think more…
Student 10:	Think about her problems so that she could fix them…
Student 3:	Rethink her model…
Jenny:	And what happened at the end?...
Student 8:	She found out that she used different things but then she went back to change it and made it right.
Jenny:	Think about that last page. Was it perfect?
Whole Class:	No!!
Jenny:	But did it do the job?
Whole Class:	Yes!!!

challenging problems. In doing so, she phrased different questions to prompt students to think about examples and non-examples of growth mindset to provide students a framework for giving constructive feedback. Jenny's questioning techniques built on students' resources to make connections between pre-existing knowledge and new lesson content. By leveraging their existing resources, she assigns value to students' experiences and draws upon their funds of knowledge (Gonzalez et al., 2006).

Discussion

Our findings on instructional practices from Jenny's classroom can be used to support and inform strategies for the integrating CT, language, and literacy instruction. While there is a growing body of work on CT-ELA integrated curricula (Bers, 2019; Burke & Kafai, 2012), little research focuses on the instructional strategies that meet the specific needs of multilingual students (see Jacob et al., 2020). Typically, what has been missing in the literature is the specification of how the heterogeneous backgrounds of multilingual students influence their learning processes. Given that students in Jenny's class come from mostly Spanish speaking families and communities, but are schooled in English, their home language proficiency levels vary. Students from heterogeneous communities such as these tend to display proficiency in oral and written genres of informal English and leverage their everyday sense-making abilities to understand complex computational concepts (NASEM, 2018). To this end, Jenny's use of verbal questioning techniques to scaffold the children's storybook enabled her students to mobilize their oral and semiotic resources to make sense of CS lesson concepts and content.

The strategic application of instructional practices was implemented in the service of building on students' existing literacy skills to teach CT (Jacob & Warschauer, 2018). This investigation stands in contrast to empirical studies focusing on how to integrate CT and ELA instruction. Emerging CT and literacy frameworks advance this discussion to situate computational thinking as a literacy in itself across multiple dimensions. However, what has been lacking is theoretical frameworks focusing on the overlap between CT, language, and literacy learning that informs instructional practices for culturally and linguistically diverse learners. The CT and literacy framework put forth by Jacob and Warschauer (2018) can be used as an analytic framework to highlight how instructional strategies mobilize the existing literacy and CT resources of students with heterogeneous linguistic needs.

Implications

Based on our findings, we suggest that practitioners apply strategies for teaching CT-ELA integrated instruction that leverages students' existing resources to foster CT, language, and literacy skills. Practitioners who integrate CT curricula with narrative genres can use students' knowledge of storytelling devices to teach CT concepts. When serving multilingual students, teachers should also be aware of students' heterogeneous backgrounds. For students who are learning English and their home language at the same time, instruction that leverages their everyday language solidifies CS knowledge in preparation for engaging students in more demanding scientific and technical language. Finally, CS content should not be taught to the exclusion of the dispositions that will enable students to develop a sense of efficacy and belonging as computer scientists. Therefore, supplementing the curriculum with instructional materials, such as children's books, about diverse pioneers in the field of CS who persevere in the face of adversity is an excellent way to foster student identification with the discipline.

Acknowledgements

We would first like to thank the teachers and students who invited us into their classrooms and supported this work. We would also like to thank the reviewers for giving this work a platform. Finally, we would like to thank the National Science Foundation (grant 1738825 and 1923136) for providing the funding that made this project possible. Findings expressed in this work are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Bers, M. U. (2019). Coding as another language: A pedagogical approach for teaching computer science in early childhood. *Journal of Computers in Education, 6*(4), 499-528.

Brennan, K., Balch, C., & Chung, M. (2014). Creative computing. Harvard Graduate School of Education. http://creativecomputing.gse.harvard.edu/guide/

Burke, Q., & Kafai, Y. B. (2012, February). The writers' workshop for youth programmers: Digital storytelling with Scratch in middle school classrooms. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 433-438).

Connolly, J. H. (2001, July). *Context in the study of human languages and computer programming languages: A comparison*. In International and Interdisciplinary Conference on Modeling and Using Context (pp. 116-128). Springer.

diSessa, A. (2000). *Changing minds: Computers, learning and literacy*. MIT Press.

Dorph, R., Shields, P., Tiffany-Morales, J., Hartry, A., & McCaffrey, T. (2011). *High hopes–few opportunities: The status of elementary science education in California*. Sacramento, CA: The Center for the Future of Teaching and Learning at WestEd.

García, O., & Wei, L. (2015). Translanguaging, bilingualism, and bilingual education. *The Handbook of Bilingual and Multilingual Education*, 223, 240.

Gomez-Zwiep, S. (2017, March). *Creating equitable STEM spaces for English learners* [Paper presentation]. California STEM Includes Conference, Anaheim, CA, United States.

González, N., Moll, L. C., & Amanti, C. (Eds.). (2006). *Funds of knowledge: Theorizing practices in households, communities, and classrooms*. Routledge.

Hsieh, H. F., & Shannon, S. E. (2005). Three approaches to qualitative content analysis. *Qualitative Health Research*, 15(9), 1277-1288.

International Society for Technology in Education (ISTE) & Computer Science Teachers Association (CSTA), (2011). Operational definition of computational thinking for K-12 education.

Jacob, S. R., & Warschauer, M. (2018). Computational thinking and literacy. *Journal of Computer Science Integration, 1*(1), 1-19.

Jacob, S., Nguyen, H., Tofel-Grehl, C., Richardson, D., & Warschauer, M. (2018). Teaching computational thinking to English learners. *NYS TESOL journal, 5*(2), 12-24.

Jacob, S., Nguyen, H., Garcia, L., Richardson, D., & Warschauer, M. (2020, March). Teaching Computational Thinking to Multilingual Students through Inquiry-based Learning. In 2020 *Research on Equity and Sustained Participation in Engineering, Computing, and Technology* (RESPECT) (Vol. 1, pp. 1-8). IEEE.

Janzen, J. (2008). Teaching English language learners in the content areas. *Review of Educational Research, 78*(4), 1010-1038.

Jona, K., Wilensky, U., Trouille, L., Horn, M. S., Orton, K., Weintrop, D., & Beheshti, E. (2014, January). Embedding computational thinking in science, technology, engineering, and math (CT-STEM). In *Future directions in computer science education summit meeting, Orlando, FL*.

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, 11*(1), 44-53.

Lee, O., & Fradd, S. H. (1998). Science for all, including students from non-English-language backgrounds. *Educational Researcher, 27*(4), 12–21.

Lee, O., Llosa, L., Grapin, S., Haas, A., & Goggins, M. (2019). Science and language integration with English learners: A conceptual framework guiding instructional materials development. *Science Education, 103*(2), 317-337.

Michaels, S., & O'Connor, C. (2015). Conceptualizing talk moves as tools: Professional development approaches for academically productive discussions. In L. B. Resnick, C. Asterhan, & S. Clarke (Eds.), *Socializing intelligence through academic talk and dialogue* (pp. 347-362).

National Academies of Sciences, Engineering, and Medicine. (2018). *English learners in STEM subjects: Transforming classrooms, schools, and lives*. The National Academies Press. doi:h10.17226/25182

National Research Council. (1996). National science education standards. *ERIC Document Reproduction Service, 391*(690). National Academy Press.

National Research Council. (2012). *A framework for K–12 science education: Practices, crosscutting concepts, and core ideas*. National Academies Press.

National Research Council. (2014). *STEM integration in K-12 education: Status, prospects, and an agenda for research*. The National Academies Press. doi:10.17226/18612

Pane, J. F., & Myers, B. A. (2001). Studying the language and structure in non-programmers' solutions to programming problems. *International Journal of Human-Computer Studies, 54*(2), 237-264.

Peppler, K. A., & Warschauer, M. (2011). Uncovering literacies, disrupting stereotypes: Examining the (dis) abilities of a child learning to computer program and read. *International Journal of Learning and Media, 3*(3), 15-41.

Proctor, C. P., Dalton, B., & Grisham, D. L. (2007). Scaffolding English language learners and struggling readers in a universal literacy environment with embedded strategy instruction and vocabulary support. *Journal of Literacy Research, 39*(1), 71-93.

Rosebery, A. S., & Warren, B. (Eds.). (2008). *Teaching science to English language learners: Building on students' strengths*. National Science Teachers Association.

Ryoo, J. J., Bulalacao, N., Kekelis, L., McLeod, E., & Henriquez, B. (2015, September). Tinkering with "failure": Equity, learning, and the iterative design process. In *FabLearn 2015 Conference at Stanford University, September 2015*.

Sengupta, P., Dickes, A., & Farris, A. (2018). Toward a phenomenology of computational thinking in STEM education. *Computational Thinking in the STEM Disciplines*, 49-72.

Shea, L. M., & Shanahan, T. B. (2011). Methods and strategies: Talk strategies. *Science and Children, 49*(3), 62-66.

Turner, E. E., & Bustillos, L. M. (2017). Qué observamos aquí? Qué preguntas tienen? Problem solving in Ms. Bustillos's second-grade bilingual classroom. In *Access and equity: Promoting high quality mathematics in PK–2* (pp. 45–63). National Council of Teachers of Mathematics.

Vee, A. (2017). *Coding literacy: How computer programming is changing writing*. MIT Press.

Vogel, S., Hoadley, C., Castillo, A. R., & Ascenzi-Moreno, L. (2020). Languages, literacies and literate programming: can we use the latest theories on how bilingual people learn to help us teach computational literacies?. *Computer Science Education, 30*(4), 420-443.

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.

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

Wright, T. S., & Gotwals, A. W. (2017). Supporting kindergartners' science talk in the context of an integrated science and disciplinary literacy curriculum. *The Elementary School Journal, 117*(3), 513-537.