UNIVERSITY OF CALIFORNIA,
IRVINE

Breaking the Language Code: Unlocking Computer Science for Multilingual Students

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Education

by

Sharin Jacob

Dissertation Committee:
Professor Mark Warschauer, Chair
Professor Emeritus Debra Richardson
Professor Penelope Collins

2022

DEDICATION

To my mother and father, Fawzia and Ernest Jacob, I dedicate this dissertation. All that is good in me has come from you. Thank you for all of the sacrifices you have made for your children. I love you from the bottom of my heart and to the ends of the earth.

TABLE OF CONTENTS

Page

LIST OF FIGURES **vi**

LIST OF TABLES **vii**

ACKNOWLEDGMENTS **viii**

VITA **x**

ABSTRACT OF THE DISSERTATION **xii**

1 Introduction **1**

2 Examining the What, Why, and How of Multilingual Student Identity Development in Computer Science **8**

2.1 Chapter Synopsis 8
2.2 Introduction 9
2.3 Background 13
2.3.1 Identity Studies in CS Education 13
2.4 Theoretical Framework 16
2.4.1 Constructing Disciplinary Identities 16
2.4.2 Multilingual Student Identity Construction 18
2.5 Research Context 19
2.5.1 Multilingual Students and CS Education 19
2.5.2 The Research Practice Partnership 20
2.5.3 Overview of the Computational Thinking Curriculum 21
2.5.4 Teacher Professional Development 26
2.6 Method 28
2.6.1 The Current Study 28
2.6.2 Data Sources 29
2.6.3 Data Analysis 31
2.7 Results 32
2.7.1 Breakdown of Student Demographics 32
2.7.2 ICSM? Survey Results 32
2.7.3 Results from Student CS Identity Interviews 36
2.8 Discussion 51

2.8.1 Students' Experiences with Computers . . . 52
2.8.2 Students' Perceptions of Computer Scientists . . . 53
2.8.3 Students' Interest in Computer Science . . . 53
2.8.4 Students' Perceptions of Themselves as Computer Scientists . . . 54
2.8.5 Students' Perceptions of Making Mistakes . . . 55
2.8.6 Support from Peers . . . 56
2.8.7 Learning CS Outside of School . . . 56
2.8.8 Limitations . . . 57
2.8.9 Conclusions . . . 59

3 Integration of Computational Thinking Into English Language Arts 61
3.1 Chapter Synopsis . . . 61
3.2 Introduction . . . 62
3.3 Theoretical Framework . . . 63
3.3.1 Computational Literacies . . . 63
3.4 The CS-ELA Integrated CT Curriculum . . . 65
3.4.1 Linguistic Scaffolding . . . 66
3.5 Methods . . . 68
3.6 Results . . . 70
3.6.1 Strategies Used for CT-ELA Integration . . . 70
3.6.2 Applying a CT and Literacy Framework Through CT-ELA Integration 71
3.7 Discussion . . . 72
3.8 Implications . . . 74

4 Teaching computational thinking to multilingual students through inquiry based learning: A cross-case analysis 76
4.1 Chapter Synopsis . . . 76
4.2 Introduction . . . 77
4.3 Theoretical Framework . . . 79
4.3.1 Inquiry-based Learning . . . 79
4.3.2 Inquiry-based Learning and Computational Thinking . . . 80
4.3.3 Types of computer science inquiry . . . 82
4.3.4 Inquiry-based Learning in STEM and Computer Science Education . 83
4.4 Overview of the Computational Thinking Curriculum . . . 84
4.5 Method . . . 85
4.5.1 The Current Study . . . 85
4.5.2 Sampling Procedures . . . 86
4.5.3 Data Collection Methods . . . 86
4.5.4 Results . . . 88
4.6 Results . . . 90
4.6.1 Open Approaches to Inquiry . . . 90
4.6.2 Structured Approaches to Inquiry . . . 92
4.7 Student Outcomes . . . 96
4.7.1 Student development of computational thinking through programming in Scratch . . . 96

4.7.2 Student identification with computer science 98
4.8 Discussion 98
4.8.1 Overview of Findings 99
4.8.2 Limitations and Future Research 101

5 Discussion **103**
5.0.1 Significance of the Research 104
5.0.2 Practical Implications 104
5.0.3 Methodological Implications 105
5.0.4 Integrating Multilingual Students' Identities, Literacies, and Pedagogies 107
5.0.5 Implications for Educational Leaders, Policymakers, Teachers, and Researchers 110
5.0.6 Conclusion 112

Bibliography **114**

LIST OF FIGURES

Page

2.1 Aggregate Pre- and Post-Survey Average Responses 33

3.1 Three-Dimensional Framework for Understanding Computational Thinking and Literacy (Jacob & Warschauer, 2018) 63
3.2 Computer Science Language Functions 68

4.1 Inquiry-based Continuum Based on the Degrees of Freedom in Conducting Investigations 83
4.2 Students Combined Scores for Complexity, Mechanics, User Experience, and Computer Science Constructs, by Class 96
4.3 Frequency of Block Usage According to Computational Thinking Constructs, by Class 97
4.4 Differences in Project Scores by Category, by Class 98
4.5 Differences in Identity Survey Scores by Category, by Class 99

5.1 Theoretical framework for understanding asset-based CS pedagogy for multilingual students 107

LIST OF TABLES

Page

2.1 Student Demographics 32
2.2 Student Interest in Computer Science 34
2.3 Community Support for Computer Science 36

3.1 Sample learning goals that integrate Grade 4 Common Core ELA, English Language Development, and Computer Science Teaching Association standards 67
3.2 Audio Transcript of Jenny Teaching The Most Magnificent Thing 73

ACKNOWLEDGMENTS

I am deeply grateful for the support and guidance of my dissertation committee. First, I want to acknowledge, with my deepest gratitude, the interest, generosity, enthusiasm, and support of my advisor and dissertation chair, Dr. Mark Warschauer. His support has been unwavering from the moment I entered the Ph.D. program as a first-year student. His plans for me have surpassed any professional or academic vision of a future that I could come up with myself. I remember walking UCI's Ring Road in my first year as a grad student and discussing our future plans, and it sounded all but impossible. Nevertheless, so far each and every goal and dream we have envisioned and discussed has come true. I would like to also thank Dr. Debra Richardson who has been so generous with her time, attention, and support over these last years. She is a force of nature and has inspired me to fight for what I believe in and align my work with my highest intentions and values. She has provided endless support for my work and my goals and has shown unwavering dedication from day one. Finally, I would like to thank Dr. Penelope Collins who has been a champion and supporter of my work since the day I stepped on campus. I remember her enthusiasm for my research on the first day I presented this work to the Digital Learning Lab. She only ever saw the best in me and made me believe in myself and my future as a scholar.

I would also like to thank Dr. Lia Kamhi-Stein, Dr. Simeon Slovacek, and Dean Cheryl Ney at Cal State LA for their endless support and encouragement. Lia believed in me even when I didn't believe in myself and gave me countless opportunities that helped me grow into the person I am today. Dr. Slovacek never let me settle and mentored me in the research and grant process that opened the door to PhD programs and beyond. Dr. Ney took me under her wing, mentored me, and taught me how to be a humble leader. Each and every time I tried to thank her, she always simply asked me to pay it forward. I will always do my best to apply that intention to everything I do. To the faculty and Dean who believed in me at Cal State LA, I will be forever grateful.

I also want to acknowledge my ECforALL project team who spent countless hours discussing ideas, providing feedback, and most importantly, working with our teachers and our kids. They worked tirelessly to create a challenging, safe, and intellectually nurturing environment, and they were always respectful of the research. The impact that they had on me and on every child with whom they worked cannot be overstated, and I am more grateful than they will ever know.

I would also like to thank Bryan Twarek and Bill Marsland who formed the bedrock of a partnership that has developed into a large-scale project that excites and provides real opportunities for young children from diverse backgrounds. Their generosity and friendship are truly valued and I would not be where I am today without their support and amazing example.

Finally and most importantly, I want to acknowledge my family, friends, and community who provided the bedrock of support without which I would never have grown into the person I am today. My husband, Ben Gillen, has been my rock and sounding board throughout my

entire Ph.D., supporting me at every level of the human experience, without whom I would not be here today. He has made countless and endless sacrifices for me that one wouldn't believe if I told them and is the epitome of who a good person is and should be. I would like to thank my brother Sharif, who has believed in me in both my happiest and most difficult hours and is one of the people in this world that I look up to the most. Every time I make a good choice, nail a presentation, write a good paper, or teach a great class, I say to myself "I still have a little Sharif in me." I would like to thank Yassin, my little brother who has shown me what love and sacrifices are truly made of, and who would never let me give up, no matter what it took. I would like to thank my mother and father who made nameless sacrifices for their children, for us to have a better life, and I hope I can only achieve a glimpse of what they have hoped for me. I dedicate all the good that I may have done in this world to them. Finally, I would like to thank Dr. Sandra Arroyo who knows where I started and how far I have come and who has wanted nothing but the best for me. She has spent countless hours of hard work and dedication to help me reach my potential and achieve my most cherished dreams. She is one of the most caring and selfless people I know. To her, I will be forever grateful.

I would also like to acknowledge all of the teachers and kids with whom we worked. They were so generous to allow us into their rooms and their lives and taught us so much about what it means to learn, grow, and persist.

I would like to thank the National Science Foundation (grants 1738825 and 1923136) for providing the funding that made this project possible. The findings expressed in this work are those of the author and do not necessarily reflect the views of the National Science Foundation.

Finally, I would like to thank the Association for Computing Machinery and the Institute of Electrical and Electronics Engineers who have generously giving me permission to incorporate my published work into my dissertation.

VITA

Sharin Jacob

1999-03	B.A. in Philosophy, San Francisco State University
2004-05	Teaching Credential, San Diego State University
2013-15	M.A. in TESOL, California State University, Los Angeles
2017-22	Ph.D. in Education, University of California, Irvine

FIELD OF STUDY

Computer Science Education

PUBLICATIONS

Jacob, S., Montoya, J., Nguyen, H., Richardson, D., & Warschauer, M. (in press). Examining the What, Why, and How of Multilingual Student Identity Development in Computer Science. ACM Transactions on Computing Education.

Jacob, S. R., Parker, M., & Warschauer, M. (in press). Integration of Computational Thinking into English Language Arts. ACM Special Issue on K-5 Computational Thinking.

Montoya, J., Jacob, S. R., & Warschauer, M. (in press). Exploring multilingual students gender identities in computer science education. Teachers College Record.

Prado, Y., Jacob, S., & Warschauer, M. (2021). Teaching computational thinking to exceptional children: Lessons from two inclusive classrooms. Computer Science Education.

Kamhi-Stein, L. D., Jacob, S. R., Herrera, A., & Seaborne, R. (2021). Linking a community based ESL program with the MA in TESOL practicum course: The tale of a program. CATESOL Journal.

Zhou, N., Chao, Y., Jacob, S., & Richardson, D. (2020). Teacher perceptions of equity in high school computer science classrooms. ACM Transactions on Computing Education, 20(3), pp. 1-27.

Jacob, S. R., Nguyen, H., Garcia, L., Richardson, D., & Warschauer, M. (2020). Teaching computational thinking to multilingual students through inquiry based learning: A crosscase analysis. Proceedings of the IEEE Annual International Conference on Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT'20).

Nguyen, H., Garcia, L., Jacob, S. R., Richardson, D., & Warschauer, M. (2020). Classroom Use of Discourse-Rich Tools to Promote Computational Thinking. Proceedings of the IEEE Annual International Conference on Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT'20).

Nguyen, H., Garcia, L., Jacob, S., & Warschauer, M. (2020). Reflection as formative assessment of computational thinking in elementary grades. Proceedings of the International Conference on the Learning Sciences (ICLS'20).

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), pp. 12-24.

Jacob, S. R., Montoya, J., & Warschauer, M. (in press). Examining identity performance of multilingual students in computer science education: An ethnographic case study. In G. Kessler (Eds.), Identity, multilingualism, and CALL. CALICO Book Series: Advances in CALL Research and Practice.

Jacob, S. R., Garcia, L., & Warschauer, M. (2020). Engaging multilingual identities in computer science education. Freiermuth, M. R. Editor & Zarrinabadi, N. Editor (Eds.), Technology and the Psychology of Second Language Learners and Users. PalgraveMacmillan. https://doi.org/10.1007/978-3-030-34212-8_12

Jacob, S., Maamuujav, U., & Warschauer, M. (2020). Online Englishes. In A. Kirkpatric (Ed.), The Routledge Handbook of World Englishes. New York: Routledge.

Jacob, S. R., Bailey, A., Bers, M. U., Burke, Q., Denner, J., Franklin, D…. Warschauer, M. (2021). Computer science for multilingual students: Report of the AERA Educational Research Conference. American Educational Research Association.

ABSTRACT OF THE DISSERTATION

Breaking the Language Code: Unlocking Computer Science for Multilingual Students

By

Sharin Jacob

Doctor of Philosophy in Education

University of California, Irvine, 2022

Professor Mark Warschauer, Chair

Computer science (CS) knowledge is becoming ever more central to successful participation in today's society. Not only is fundamental computational literacy necessary to take advantage of growing employment opportunities demanded by the information economy, but computer literacy is becoming increasingly integrated into most professional roles. All aspects of modern life, from how we communicate to how we navigate, increasingly revolve around computational devices. Increasing accessibility to CS programs is a critical component for bringing underrepresented groups into the information-technology revolution. This is especially important for the large and growing population of students designated as English learners (ELs) – which grew from 8.1% in 2000 to 9.6% in 2016, and is projected to reach 25% of the student population by 2025. Compared with other groups, there is an alarming underrepresentation of ELs in CS education. In fact, high schools with 12% or more students designated as English learners offer half as many CS courses as other schools. Without engaging this population in CS, ELs are at risk for being left behind in the new information economy.

My research project is one of the first initiatives in the nation to develop and test a CS curriculum that helps predominantly Latinx, low-income, English Learners succeed in computing. This curriculum employs what we know about language, culturally relevant pedagogy,

and instructional design to help ELs and other diverse students leverage their own and their community resources for greater success and entry into the field of computing.

My first dissertation study examines how and why multilingual students identify with CS through their participation in the curriculum. I administered a pre- and post-CS identity survey to students (n= 108) in seven classrooms. To better understand trends in students' responses, I conducted semi-structured interviews of four students in each of the classrooms (n=28). Findings indicated that tailored instruction provides opportunities for connections to out-of-school learning environments with friends and family that may shift students' perceptions of their own abilities to pursue computer science and persist when encountering challenges.

My second study examines how teachers integrate language and literacy into CS curricula to improve language and content skills for multilingual students. I performed qualitative analysis on field notes involving top-down and bottom-up coding that starts with categories from prior research on teaching STEM to ELs and refines that coding based on emergent themes from this study. 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 CS concepts and practices.

The third study entails an exploratory cross-case study of five teachers' classrooms. Videotaped lessons and classroom observations identified the modes of instruction and interaction in each classroom. Analysis of pre- and post-CS identity surveys and computer science assessments measured how students performed in each of the classrooms. Findings indicated that more structured approaches to inquiry-based instruction appeared to support the development of computational thinking skills and CS identities for multilingual students.

This dissertation represents one of the first major research efforts aimed at systematically investigating and identifying promising practices for teaching Latinx, low-income, English

learners in CS. It contributes to our knowledge about culturally responsive CS instruction, the intersection between CS education and language and literacy development, and effective approaches for teaching CS in diverse elementary schools. The aim of this research is to initiate a new line of inquiry concerning quality CS instruction for multilingual students in the US.

Chapter 1

Introduction

The importance of computer science (CS) knowledge, skills, and attitudes is becoming ever more central to successful participation in today's society. Not only is fundamental computational literacy necessary to take advantage of growing employment opportunities demanded by the information economy, but computer literacy is becoming increasingly integrated into most professional roles. All aspects of modern life, from how we communicate to how we navigate, increasingly revolve around computational devices. A critical task central to this information-technology revolution in our society involves broadening participation in computer science.

Broadening participation in computer science is especially important for the large and growing population of students designated as English learners (ELs) – which grew from 8.1% in 2000 to 9.6% in 2016, and is projected to reach 25% of the student population in 2025 [126]. Compared with other groups, there is an alarming underrepresentation of students designated as ELs in CS education and achievement. In fact, high schools with 12% or more students designated as English learners offer half as many CS courses as other schools [125]. Without engaging this population in CS, they are at risk for being left behind in the new

information economy.

Engaging students designated as Els in computer science instruction brings its own particular linguistic challenges and opportunities. Successfully engaging these students in computer science requires developing literacy skills due to discipline-specific discourse structures and demanding technical language [95]. Through the implementation of a computational thinking curriculum that integrates intensive and targeted language support, we found that ELs students improved their language and literacy skills while learning computer science [138].

Over the last few years, researchers have increasingly recognized the need to address cultural factors that adversely affect the multilingual student participation in CS. Multilingual students face key disadvantages in learning CS, including fewer course offerings in CS [125], reduced access to home computers or family members who are knowledgeable about CS [168], lack of diverse role models in CS whether through direct experience or through media representations [202], and decontextualized and individualized methods of CS instruction that are not a good match for the cultural values of culturally and linguistically students and families [29]. After integrating culturally responsive materials into our curriculum, we found that connecting to students' cultural resources and communities improved students' perceived ability to pursue computer science and willingness to persist when encountering challenges [93].

Beyond cultural factors, CS instructional practices for multilingual students require further study and development. Research indicates that diverse students benefit from more structured approaches to computer science instruction [94]. We found the integration of structured approaches to scientific inquiry appeared to support multilingual students' development of computational thinking skills and computer science identities [94].

While there are a number of promising initiatives geared toward improving access to quality CS education in high school for diverse learners [78], by this age, wide disparities already

exist in diverse groups' knowledge, skills, and attitudes toward STEM [37], making it difficult for those on the margins to catch up [181]. This is especially true in CS, since low-SES learners typically lack access to the kinds of out-of school experiences through robotics clubs, coding summer camps, and at-home mentoring and modeling that help promote early CS knowledge and identity development [13, 114, 202]. Finally, access to quality CS education for multilingual students, especially in elementary school programs, is highly complex and challenging. First, the elementary school curriculum is already packed and test-driven [211], incentivizing teachers to prioritize tested subjects, such as English and math, rather than non-tested subjects, such as CS. Additionally, only a minority of elementary school teachers themselves have the CS content and pedagogical knowledge to confidently introduce the subject to their students [162].

Within the growing body of research on broadening participation in CS, adding specific research on serving multilingual students in CS will provide a roadmap for educators serving this population in improving CS knowledge, skills, and attitudes. This dissertation examines the integration of linguistically and culturally responsive materials when designing CS instructional curricula to mobilize multilingual students' linguistic, cultural, and conceptual resources in encouraging their engagement in highly complex and demanding CS content.

While there has been a growing body of research on two related fields – (a) English learners in STEM and (b) CS for all – there is almost no specific research on English learners in CS, thus providing no roadmap for educators as to how to improve CS knowledge, skills, and attitudes for this population. Building on scholarship on culturally sustaining pedagogy [147, 148], we seek to examine how CS instructional materials can integrate evidence-based practices that leverage multilingual students' conceptual, linguistic, and cultural resources to engage students in highly complex and demanding CS content. Engaging multilingual students in computer science education offers additional affordances considering the relationship between learning programming languages and learning a new language. Siegmund

et al. [182] use fMRI data to study the learning of programming languages in introductory programming courses, and finds that programming activates natural language processing areas of the brain, as opposed to math or logic. Peppler and Warschauer (2011) examined the affordances of creative programming in media rich environments such as Scratch to provide emergent literacy development. Through case study analysis, they find that Brandy, a child with intellectual disabilities, was able to leverage her programming abilities during participation in an after-school computer clubhouse to develop metalinguistic awareness, reading and writing skills, self-efficacy, and a sense of identity as a computer scientist [155]. Similarly, Peppler [153] examines how youth engage in literacy practices through media arts construction, finding that programming in media rich environments such as Scratch extend beyond traditional definitions of print literacy, computer literacy (i.e., typing, word processing) and media education (i.e., taking critical stances towards texts). Through the combination of different media (i.e., audio, animated, written, and kinesthetic), youth engaged in complex forms of multimodal communication that extend beyond traditional forms of literacy to encompass media literacy, technological fluency, and artistic expression. In the same vein, Jacob and Warschauer [96] propose a three dimensional framework for exploring the relationship between computational thinking and literacy through: 1) situating computational thinking in the literature as a literacy; 2) outlining mechanisms by which students' existing literacy skills can be leveraged to foster computational thinking; and 3) elaborating ways in which computational thinking skills facilitate literacy development. This growing body of work calls for further exploration of the affordances computer programming poses for literacy development.

Prior work has found that there are three ways for increasing diverse learners' interest in CS: 1) leveraging student identity development [99, 170, 196, 97], 2) integrating evidence based practices into computational thinking curricula [52, 80], and 3) providing inquiry-based approaches to learning [201, 15]). However, this work has not looked at in-school contexts with large numbers of multilingual students. Therefore, this dissertation proposes to inves-

tigate these three issues in further depth. My dissertation would attempt to address a few specific questions regarding how a computational thinking curriculum is systematically implemented to develop predominately Latinx, low-socioeconomic status, multilingual students' computational thinking skills and computer science identities. This research represents the exploratory phase of a larger project intended to refine, test, and scale the curriculum, first to three participating school districts, and then nationally. For the purpose of this proposal, all data has been collected according to the data collection plan for implementation of the curriculum in the 2018-2019 academic year. In the following three paragraphs, I will briefly describe each of the three studies.

In the first study, I investigate the kinds of identity work multilingual students engage in after participating in a computational thinking curriculum designed to meet their linguistic and sociocultural needs. Unlike math and science, computer science is not being implemented systematically in K-12 schools. This implies that students need early exposure to computer science in order to develop interest in the discipline and persist in pursuing CS related careers. In my study, I investigate whether early exposure to a computational thinking curriculum that has been adapted to meet the needs of culturally and linguistically diverse students positively impacts student identification with computer science. I implement an explanatory sequential mixed-methods design using a pre-and post-survey intended to measure student identification with computer science, and follow up with semi-structured interviews that further probe students' responses. These findings can be used to advance our understanding of how and why multilingual students develop discipline-specific identities through their engagement in a responsive curriculum.

The second study 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 theoretical 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.

In the third study, I turn to teacher instructional practices to explore how different approaches to inquiry-based learning appeared to support or constrain students' development of computational thinking skills and computer science identities. I adopted a cross-case mixed-methods design to collect data from five teachers and 149 students including detailed field notes, teacher interviews, student computational artifacts, and student identity surveys. Through analyses of teacher moves, preliminary findings indicated that teachers adopted different approaches to inquiry that can be indexed along a continuum ranging from open to closed. Patterns in student data revealed that those who received more structured inquiry lessons developed more sophisticated computational artifacts and showed greater identification with the field of computer science. Findings from this study are being used to add more structured inquiry approaches to the next iteration of our curriculum, including integrating USE/MODIFY/CREATE [113, 205] models into lessons and applying metacognitive strategies from reading research to students' programming activities. I intend to conduct further analysis on this study by analyzing teacher interviews from classrooms that provided more structured approaches to uncover the challenges and opportunities afforded by such approaches.

My dissertation represents one of the first major research efforts aimed at systematically

investigating and identifying promising practices for teaching Latinx, low-income, students designated as English learners in CS. It contributes to our knowledge about culturally responsive CS instruction, the intersection between CS education and language and literacy development, and effective approaches for teaching CS in diverse elementary schools. The aim of this research is to initiate a new line of inquiry concerning quality CS instruction for multilingual students in the US.

We will now turn to chapter two, which presents the first study in this dissertation examining the factors that contribute to computer science identity development in multilingual students. Next we move to chapter three, which presents the second study of this dissertation exploring the relationship between computational thinking and literacy development, specifically investigating how teachers can leverage students' existing literacy skills to develop their computational thinking skills. Then we go on to chapter four, which presents the third study of this dissertation investigating the types of inquiry-based instruction that support multilingual students' development of computational thinking skills and computer science identities. Finally we conclude with a discussion of the broader significance of this research project, discussing both the practical and methodological implications of this work.

Chapter 2

Examining the What, Why, and How of Multilingual Student Identity Development in Computer Science

2.1 Chapter Synopsis

Developing student interest is critical to supporting student learning in computer science. Research indicates that student interest is a key predictor of persistence and achievement. While there is a growing body of work on developing computing identities for diverse students, little research focuses on early exposure to develop multilingual students' interest in computing. These students represent one of the fastest growing populations in the US, yet they are dramatically underrepresented in computer science education. This chapter examines identity development of upper elementary multilingual students as they engage in a year-long computational thinking curriculum, and follows their engagement across multiple settings (i.e., school, club, home, community). Findings from pre- and -post surveys of

identity showed signiicant diferences favoring students' experiences with computer science, their perceptions of computer science, their perceptions of themselves as computer scientists, and their family support for computer science. Findings from follow-up interviews and prior research suggest that tailored instruction provides opportunities for connections to out-of-school learning environments with friends and family that may shift students' perceptions of their abilities to pursue computer science and persist when encountering challenges.

2.2 Introduction

It is well established that there is a strong relationship between students' perceptions of themselves as professionals in Science, Technology, Engineering, and Mathematics (STEM) and their career interest [10, 12, 88, 141], indicating the importance of developing young peoples' attitudes and interests early on. Furthermore, student interest is a key predictor of persistence and achievement [89]. Thus, it is critical that we develop strategies for broadening the participation of students from diverse backgrounds who identify with the discipline of computer science (CS).

CS knowledge, skills, and attitudes are becoming increasingly necessary for full participation in today's society [209]. The US Bureau of Labor Statistics predicts there will be 1.4 million CS job openings by 2026 with only 500,000 qualified graduates to fill these positions [142]. In response to this overwhelming need, the White House's Computer Science for All (CSforAll) initiative has emerged over the last few years to help all students become developers, not just consumers of technology [185]. In order to broaden participation in computing, a critical task is leveraging the wealth of talent in this nation's diverse population. This is especially important for the large and growing population of students designated as English learners. This population grew from 8.1 % (or 3.8 million students) in 2000 to 9.6% (or 4.9 million students) in 2016 and it is projected to reach 25% of the student population in 2025 [126],

but it is seriously underrepresented in education and achievement [125]. Leveraging these students' linguistic and cultural diversity enables new perspectives that foster creative and innovative approaches to problem solving. Such new perspectives are not only needed to drive future technological advances; they are also becoming increasingly critical in solving pressing problems across essential domains of the human experience [200, 130].

Unfortunately, a number of factors explain the chronic lack of representation of linguistically diverse students. First, there is a dearth of data about students designated as English learners (ELs) in CS courses. As schools base their decisions on data [173], this means that educators and stakeholders must rely on assumptions about the success of students designated as ELs in CS, or that they ignore these subgroups entirely. To exacerbate this issue, schools with 12% or greater students designated as English learners offer half as many CS courses as other schools [125]. In addition to lack of access, pervasive stereotyping in the field is perpetuated through media representation. Only about 16% of students report seeing computer scientists who look like them in the media, which sends messages to diverse learners about who does CS [40]. Finally, much of the CS curricula implemented to date do not reflect the traditions and values of culturally and linguistically diverse students and their families [124]. Purposefully tailored instruction can address these issues by providing opportunities for connections to out-of-school learning environments, in which students are with friends, family, and community, which may shift students' perceptions of their abilities to pursue CS and empower them to persist when encountering challenges.

Issues regarding the classification of English learners only exacerbate inequities for these students. These students are commonly referred to as English learners because in many states the English language proficiency exam (ELP) represents the only criterion for school and districtwide designations of students' language proficiency. Other oftenly used criteria to designate students as English learners include standardized test scores in Mathematics or English Language Arts, teacher assessment of progress, and parental approval [27]. Unfortu-

nately, these are the only factors that many computer science and STEM educators consider when studying approaches to engaging these students in the classroom. Examining students in terms of their language proficiency fails to account for the linguistic and cultural assets that these students bring to the classroom [119, 199].

There are several reasons why the terms English Learner and English Language Learner are problematic [119]. First, they inherently devalue the cultural and linguistic resources that students bring to the classroom. Second, they do not reflect cultural diversity of students with these labels, as the focus is on learning English and not on the languages that these students already speak. Third, the apparent neutrality of the term renders them free from scrutiny. Fourth, they confuse students who speak English at home with those who are learning English for academic purposes, therefore delegitimizing the existing linguistic resources students can leverage for making sense of complex phenomena. Finally, they do not make room for bilingualism in instruction.

Flores and Rosa [71] argue that designations of English learners, heritage learners, and Standard English learners are constrained by raciolinguistic ideologies that lead to deficit based views of these students. They call for the denaturalization of traditional linguistic categories by reimagining education policy in order to focus on the wealth of resources imbued by these students. Building on their work, we argue that multilingualism and plurilingualism facilitate learning [73], and that multilingual students have an advantage over their monolingual counterparts in that learning involves the growth of both linguistic and subject matter competencies [157]. Multilingual students' identities are assets that can be leveraged to facilitate learning, and contribute to these students' active participation in the field of CS [93]. Therefore, we will use the term multilingual throughout this chapter to refer to students who use more than one language in their daily lives and may be learning English in school.

It is important to underscore the diversity of multilingual students, especially regarding their cultural backgrounds, languages spoken, immigration status, and time residing in the US.

Some students may be refugees whose previous instruction in their home languages and/or English may have been disrupted. Most of the students in this study are residents of the southwestern state of California and children of immigrants who arrived in the United States as young children. These children speak Spanish at home and are formally schooled in the US in English [128]. Given their diversity, it is important to avoid overgeneralizations about multilingual students as a whole; nevertheless, these students bring a shared positionality within educational institutions with regard to cultural and linguistic diversity.

In this chapter, we investigate whether early exposure to a computational thinking curriculum that has been adapted to meet the needs of culturally and linguistically diverse students appears to support identification with CS. We implement a pre- and post-survey design intended to measure multilingual students' identification with CS. The survey is followed up by interviews that probe further into the why and how of this identification. These findings can potentially advance our understanding of how multilingual students develop discipline-specific CS identities through their engagement in a well-tailored curriculum.

Our study asks the following research questions:

1. To what extent do multilingual students' CS identities change after participating in a year-long computational thinking curriculum designed to meet their linguistic and sociocultural needs?

2. What factors contribute to these changes? How are multilingual students' identities shaped by participation in both formal and informal learning environments?

2.3 Background

2.3.1 Identity Studies in CS Education

There is a nascent but growing body of work examining equitable CS education for multilingual and racially diverse students that seeks to advance our knowledge of how curriculum and practice can be designed to leverage the varied resources of these students and their communities. In a review of previous works, student identification with CS was examined through the construction of learning environments to support identity growth [178]. A recent study investigated how students from marginalized backgrounds reshaped dominant narratives, in particular by integrating quilting with paper circuitry [156]. Students who quilted with paper circuits were able to leverage their rich cultural traditions to reframe their participation in computing, utilizing their computing identities in novel ways. Similarly, there have been several efforts to draw upon existing cultural practices and traditions in students' families to reveal the mathematical and computational concepts they already use. Existing traditions in indigenous communities, such as sewing, weaving, and decorative beading, have been connected to engineering and computing practices through the utilization of electronic textiles (e-textiles) [100]. E-textiles blend crafting practices with microprocessors, light bulbs, and sensors to explore science and engineering principles, such as electricity and circuits, in culturally responsive ways.

Howell et al.[90] piloted an e-textiles science unit in a classroom containing 30% students designated as English learners. Researchers observed a marked increase in these students' participation compared to classes that taught electricity and circuits through traditional means. Reasons for this increased participation included the following: 1) students and families valued the work completed by hand and that drew upon students' conceptual resources; 2) students took their work home and received homework assistance from family members who drew on their crafting expertise; 3) students developed a collaborative environment in

which they were able to share with their peers their expertise on how to overcome various obstacles; and 4) students were able to choose and personalize their projects which led to greater identification with their creations.

Another mechanism for increasing interest in computing involved integrating computational concepts with K-12 subject areas. Computing has been integrated into a Science Technology Engineering Arts and Mathematics (STEAM) curriculum to encourage Black girls to develop wearable technologies that they could display during their dance choreographies [7]. Through this curriculum, girls were able to leverage their interests to develop CS identities while physically embodying computing concepts through creative expression. Another study, a year long upper elementary CS-integrated computational thinking and literacy curriculum, aligned with the grade 3-5 English Language Development Standards and Common Core State Standards for English Language Arts, was developed to increase both CS learning and literacy development among predominately Latinx, multilingual students [95]. This curriculum helped to facilitate these students' expression and creativity, and when combined with structured instructional practices, it appeared to support their development of computational thinking skills and overall identification with the field of CS [93, 94].

Other work has focused on bridging the gap between formal and informal CS learning environments by leveraging multilingual students' existing resources to enhance their engagement with CS. The use of translanguaging by middle school teachers has been studied to determine how it facilitates CS learning [199]. Drawing on the theory of translanguaging [74], Vogel et al. [199] situate coding as a discourse that is embedded within historical, cultural, and social contexts, arguing that educators and policy makers need to bring underserved groups such as emergent bilinguals into this discourse. Translanguaging represents a mechanism for leveraging multilingual learners' full linguistic repertoires as they engage in computational literacies. In addition to leveraging linguistic resources, researchers have implemented strategies for engaging families in the STEM education of children from predominately Latinx backgrounds

through community-based engagement efforts [56], such as hosting after-school Spanish family coding nights [51]. Students' individual and collective interests have also been leveraged in support of CS learning, such as building on Black boys' interests in video games to engage them in looking "under the hood" at the underlying technology, which led to their increased perceptions of peers as resources as well as greater access to technical literacies [97].

Challenges with material notwithstanding, diverse students from marginalized groups may lose interest in science classes very early in their academic careers, especially when the subject is presented as unrelated to their own lives and contexts [64]. Successful intervention for these students consists of making science relevant by drawing from students' own contexts and funds of knowledge [17]. This approach acknowledges the value that students of all backgrounds bring to the materials with which they engage, and validates these identities beyond mere teaching of the scientific model of thought. Additionally, students from predominantly Latinx cultures often favor relational learning over independent, noncollaborative approaches [166]. It is unsurprising that students from underrepresented groups often become more disengaged from school in upper elementary and middle grades—the disinterest of a previously engaged student develops alongside their experiences of science as noncollaborative or competitive, the product of a culture to which they do not belong [14]. Culturally responsive teaching approaches value interdependence and collaboration, and better prepares all students for the actual demands of creative thought and collaborative work in later CS careers [3]. In practice, this type of instruction provides opportunities for peer interaction and collaborative learning [33], contextualizes lessons and projects in the experiences and skills of students' homes and communities [127], and relies on an awareness of how culture and identity inform student interests in pursuing science [37]. For Latinx students in particular, educators can nurture their interest in CS by increasing their exposure to role models, personally meaningful coursework, and instruction that meets their cultural and linguistic needs [41]. Finally, for Latinx students who are also learning English as a second language during K–5 grades, instruction must meet learners' needs for scaffolded language learning, so

that they acquire the language of science along with its methods; truly responsive teaching must be both culturally and linguistically sensitive [94].

These findings establish the following factors that contribute to students' increased identification with computing: 1) reshaping dominant narratives of who does CS, 2) making cross-curricular connections, 3) providing culturally responsive teaching, and 4) leveraging their existing resources to engage in CS content. Despite the many common themes characterizing multilingual students' development of CS identities, overgeneralizations regarding this student population should be avoided, as these students display substantial variation with regard to factors including but not limited to language, culture, immigration status, and previous schooling. What follows is our theoretical framing of disciplinary identification, as experienced in particular by multilingual students.

2.4 Theoretical Framework

2.4.1 Constructing Disciplinary Identities

Identity has recently gained traction as an essential educational construct that provides a framework for understanding student learning in STEM. Identity studies have traditionally focused on the types of knowledge and expertise necessary to gain entrance into communities of practice [26, 35], but there is a growing body of work that situates identity as negotiated and contingent upon broader social, cultural, and historical contexts [34]. In this chapter, we draw from social practice theory to characterize identity as a dense network of relationships and interactions organized around structured activities [164]. These relationships are embedded in powerful narratives about what it means to be a competent actor within sociocultural and historical contexts (e.g., a computer scientist, a good student, and a person from a specific racial, cultural, or linguistic background) [34]. Therefore, as students po-

sition themselves within educational settings, their actions are constrained by the socially constructed norms, rules, and expectations that govern these spaces. A students' positionality is not static: it is dependent upon how it is taken up over time by the actors residing in these spaces, and it can be directed toward or against the norms governing these spaces.

The notion of identity can be used to better understand student learning as it is constructed through engagement or disengagement in disciplinary practices [97, 197]. Our framing of this issue borrows heavily from DiSalvo et al. [97]. Social norms and practices represent one mechanism for governing student disciplinary identification. Students are more likely to identify with a discipline if they consider its practices to be typical of their social sphere [97, 188]. Conversely, students will disidentify with a discipline if they perceive its practices to be socially atypical. Research indicates that disciplinary disidentification leads to decreased academic achievement, downward economic mobility, and social inequality [67, 187, 188]. It can also spread through peer groups by way of peer pressure [187]. Due to its virulence, disidentification represents a key contributor to underrepresentation of marginalized students, such as women, Black students, and students of color [103]. It thus becomes key to distinguish incidences when students "choose not to learn" from those in which they have difficulty grappling with the material, when examining the effects of disidentification on student learning [107].

As identity is constructed through practice, it involves acquiring the knowledge, skills, and attitudes necessary to be perceived as a competent actor within a given discipline. Identity and learning are inextricably linked to learner participation in disciplinary practices across contextualized settings [17, 36, 137, 164]. Our theoretical framework relating these two concepts, learning and identity, borrows heavily from Van Horne et al. [197]. Learning occurs in space and time across multiple contexts (in school, out of school, and at home) [11, 18, 152, 151] and when it is socially relevant [38, 83]. Providing these conditions helps to democratize knowledge and access to socially valued practices [134], thereby fostering

identity development through meaningful interaction [43, 136]. As students experience CS across time and multiple spaces, they begin to construe what it means to be part of an imagined community and how they themselves fit within these communities [53, 139].

During identity construction, there is a dynamic interplay between student perceptions of who they are and who they desire to be, which is often guided by their perceptions of what competent actors are like with a given field [34]. These perceptions are influenced by how students' actions are taken up and received by socially recognized members of the community. Therefore, identity development encompasses how students view themselves in relation to others, through their identification with disciplinary practices, as they participate in communities of practice [13]. Curricula often provide an implicit guide for how learners might engage with a given discipline. Thus, curricula that connect formal learning environments to out-of-school contexts strengthen the development of disciplinary identities [131].

2.4.2 Multilingual Student Identity Construction

Crump [50] views language and identity as inextricably linked. Both are dynamic and fluid entities, and both represent devices by which participants become members of academic communities [53]. Currently, educational policy makers distinguish between native and non-native speakers of English, and the designation of English learner only compounds the extent to which students are labeled according to what they still have to learn, as opposed to what they already know. By using the label multilingual instead, we acknowledge the multiple competencies that these students bring to bear [42], and the duality of cultures and worlds that they inhabit. First and second (or third languages) are used sequentially, to indicate that the first language can be used in service of developing the second language and content knowledge, and vice versa [4].

Furthermore, not only are multilingual students linguistically diverse, but they also reside

in sociocultural contexts that include cultural, familial, and social resources that can be leveraged to foster learning and encourage the development of disciplinary identities. Elementary English learners come from diverse communities that each have their own goals, values, and particular ways of evaluating and interacting with the world. Before entering school, children are socialized into the language of their homes and communities [84]. Over time, they encounter new cultural contexts and practices that are distinct from their communities of origin. Students leverage their everyday sense-making abilities learned from their homes and communities to interpret scientific phenomena they encounter in the classroom [117, 122, 121]. Through this sense making of scientific practices, such as identifying problems and designing solutions, students develop deeper understanding of scientific principles.

Instructional practices that promote inquiry-based, content-first approaches apply inductive methods of learning that build on students' existing resources by allowing them to use their everyday sense-making abilities to access content [28]. As students negotiate meaning in STEM classrooms, they make connections between teacher output and scientific artifacts, while problematizing knowledge and questioning misconceptions [75]. Bridging the gap between home and school learning increases their identification with academic curricula and discipline-specific content [132]. Instruction that draws on students' existing resources presents a particularly useful approach for engaging multilingual students in STEM education [27].

2.5 Research Context

2.5.1 Multilingual Students and CS Education

Several social, cultural, and linguistic factors contribute to the marginalization of multilingual students in CS education. First, these students come from socioculturally and lin-

guistically heterogeneous backgrounds that are poorly understood. For instance, students are better able to leverage their first language in service of learning when they come from linguistically homogeneous communities [27]. Students from linguistically heterogeneous communities, however, may focus on salient aspects of variation among linguistic registers in both written and oral communication to facilitate STEM learning [27]. The students participating in our study come from communities and families that predominantly speak Spanish, but they are formally schooled in English in US schools. Therefore, these students tend to pay special attention to oral and written genres of informal English and to use their everyday sense-making abilities to interpret abstract concepts.

2.5.2 The Research Practice Partnership

The purpose of this study is to examine the identity development of upper elementary multilingual students through their participation in a yearlong computational thinking curriculum designed to meet their cultural and linguistic needs. We follow student participation across multiple settings (including school, club, home, and community). The context for this study is a research practice partnership (RPP), the goals of which are to address core problems of practice facing a diverse school district implementing CS for All initiatives and to determine how to meet the needs of the district's multilingual students. The University of California, Irvine is partnering with the Orange County Department of Education and the Santa Ana Unified School District (SAUSD) to form a collaborative network of university and K–12 researchers and practitioners with the aim of promoting computational thinking for students in grades 3–5. This network functions through principles of Design-Based Implementation Research (DBIR), designing instructional materials to implement, study, and refine alongside the county and district.

SAUSD, among US school districts, has among the highest percentages 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). In Santa Ana, 71% of foreign-born individuals are from Mexico; 8% are from Asia; and 88% of Hispanics are from Mexico. Consistent with the broader Santa Ana community, the strong majority of the students in the participating district are of Mexican heritage, but smaller numbers of students are primarily from El Salvador, Guatemala, and Honduras. This district is seeking to improve student academic achievement and interest in STEM through programs that support instructor innovation and emphasize integration of STEM and English language arts curricula.

2.5.3 Overview of the Computational Thinking Curriculum

Researchers worked collaboratively with teachers to adapt an existing grade 3–5 curriculum created by a path-breaking initiative that seeks to normalize CS education in a large urban school district from PreK through 12th grade. The curriculum was deemed well-suited to our research purposes as it aligns with the Computer Science Teachers Association K-12 Computer Science Standards and emphasizes the teaching and learning of computational thinking. The curriculum was adapted to meet the needs of the district's culturally and linguistically diverse students. Design-based implementation research was used to bridge theory and practice in the design of the instructional materials. The theoretical underpinnings of curriculum design were grounded in effective practices for engaging multilingual students in STEM, as outlined in a recent report of the National Academies of Sciences [27]. According to this report, the following findings have been shown to be effective in increasing academic and social outcomes for multilingual students in STEM: 1) engaging students in disciplinary practices, 2) encouraging rich classroom discourse, 3) building on students' multiple meaning-making resources, 4) encouraging students to use multiple registers and modalities, and 5) providing explicit focus on how language functions in the discipline. Given

the paucity of empirical evidence supporting the engagement of multilingual students in CS education, we worked collaboratively with teachers and administrators during a weeklong summer institute to develop a curriculum based on the findings of this report, while tailoring materials to meet the needs of the district's diverse learners. This was achieved by 1) aligning the curriculum with CS and literacy standards and integrating inquiry-based approaches, 2) providing multiple opportunities for collaboration, 3) providing culturally responsive pedagogy and materials, 4) presenting multimodal options for learning, and 5) providing intensive linguistic scaffolding. What follows is an explanation of how our curricular adaptations align with effective practices for teaching STEM to multilingual students.

First, the curriculum integrates CS and English Language Arts tasks to **engage students in disciplinary practices** through inquiry-based exploration, modification, and creation of products. Research indicates that STEM instruction is best provided when instructional practices leverage multilingual students' cultural and linguistic backgrounds [23, 66]. To integrate CS inquiry-based approaches into the curriculum, we utilized the "5 E" model of inquiry to guide unit development: Engage, Explore, Explain, Elaborate, and Evaluate. Throughout each phase, the teacher facilitated meaningful peer-to-peer discourse about students' problem-solving processes. In this way, inquiry-based learning provides authentic contexts for language that leverage multilingual students' existing resources while making instruction more engaging for them [98, 45, 167].

Second, the curriculum **encourages rich classroom discourse** through explicit suggestions of activity formats (e.g., individual thinking time, pair programming, small group, whole class) that engage students in using disciplinary language in multiple contexts. This provides opportunities for collaboration that have been shown to facilitate learning for predominantly Latinx communities [166]. Furthermore, the professional development associated with the curriculum focused on teachers noticing students' discourse to facilitate productive talk [180].

Third, strategies that teachers use to **build on students' existing resources** to acquire

disciplinary language and CS were highlighted through unplugged activities. These activities built on students' everyday knowledge and semiotic resources while leveraging their own ways of explaining CS concepts. During monthly professional development sessions, we included teacher tips for teacher "talk moves" [129], such as asking for clarification and leveraging students' own ways of explaining to guide them toward more formal language and advanced CS concepts. Furthermore, culturally responsive childrens' story books depicting diverse characters who were pioneers in CS and engineering fields were selected to make the content relatable to students. While racial and cultural stereotypes can have negative effects on the STEM career aspirations of underrepresented youth [140], research has also demonstrated that same-race characters can have beneficial effects on Latinx students' positive dispositions and attitudes toward CS [195].

Fourth, visualizations and physical, unplugged activities were 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 vocabulary and computational thinking concepts [116]. For example, students learned about the concept and term "parallelism" first through an activity encompassing body movement and then through visualizations. This approach lies at the heart of translanguaging, as students are able to leverage their full meaning-making repertoires to engage in CS learning [199].

Fifth, the curriculum provides **explicit focus on how language functions in the discipline** by integrating language frames, which teachers made available for use by students during project reflections, peer feedback, pair programming, and help-seeking. The development of linguistic frames is grounded in systemic-functional linguistic theory [86], which proposes that language is an inherently social phenomenon in which communication serves to operationalize the syntactic and formal structures embedded in the language of a given discipline. Culturally sustaining pedagogies have been merged with systemic functional linguistics

[176] to develop students' metalinguistic awareness of contextualized language use while giving students the agency to provide meaningful critique of how knowledge is constructed. To implement linguistic scaffolding, sentence frames were printed on student-friendly placemats for each of the lessons; these were used during reflection activities to reinforce concepts and provide guided language instruction. We were careful not to provide the frames during open conversation so that students had the agency to make their own rhetorical choices, while engaging in authentic language use arising from peer-to-peer interaction. This strategy is corroborated by recent research on the affordances and challenges involved in using sentence frames and when best to employ them so that they do not stifle communication [82].

Finally, the curriculum integrates computational thinking and literacy instruction by aligning the materials with English Language Arts classes. This alignment was achieved by leveraging the affordances of media-rich programming environments such as Scratch to teach coding and decoding block-based commands and projects in the narrative and informative textual genres. Furthermore, English Language Arts lessons were developed around our culturally responsive stories to teach narrative genres, while fostering dispositions such as perseverance and iteration that are integral to the design process. Finally, CS disciplinary activities and learning goals were also aligned with standards so as to guide teachers and set clear expectations for students: researchers and teachers aligned materials with the Common Core State Standards for English Language Arts (ELA), and the statewide Department of Education English Language Development (ELD) Standards.

These strategies and principles were employed to promote students' inclusion in the CS community of practice. We aimed to develop students' CS knowledge and language through interaction and regular participation in profession-like practices and activities. Perhaps the most important contribution of our curriculum was its "content-first" approach to learning: students were encouraged to access the discipline before engaging in linguistic tasks, which might otherwise have unnecessarily averted resources away from learning CS [116]. It is

through content learning that students interact and communicate with one another and begin to co-construct knowledge, thus becoming active creators of disciplinary practices. The goal is for students to learn as scientists would, by developing understanding of complex phenomena and then creating the most efficient terms for describing what they observe. The practices presented in this curriculum promote understanding and learning as the essential goals of real-life disciplinary practice.

This content-first approach relied on two principles of practice. First, the curriculum was taught inductively through an inquiry-based approach. For example, if students are supposed to learn the concept of loops, first they might do a dance involving repeating moves, mapping out the sequence of repeating moves and actually embodying the concept, before the term "loop" is presented. Then, after leveraging their multiple resources to understand the concept before the term, the teacher would show what loops look like in Scratch. With this knowledge, students would then practice programming projects that require loops, like the "Build a Band" project in which students use loops to create a musical concert. Second, the block-based, media-rich programming environment Scratch [161] holds real promise for authentic disciplinary CS learning. Unlike more abstract programming environments, the language used during interactive programming in Scratch is often colloquial, such as "drag this block here." Herein lies the real power of tools such as Scratch: they empower students to access the discipline and develop computational thinking without having to complete challenging linguistic tasks that may distract them from the primary goal of developing understanding of CS content. It was not until students had completed the projects and shared them with their peers for feedback that the teachers presented the linguistic scaffolding to explicitly teach CS language functions and scaffold their disciplinary language.

2.5.4 Teacher Professional Development

To support the piloting teachers, the project included professional development (PD) throughout the year during team meetings and classroom debriefs, as well as during the summer institute. The PD structure was designed to engage teachers in the types of inquiry-based models fundamental to CS instruction; it followed practices of effective learning and teaching in STEM and CS education to multilingual students [27], and was then tailored based on suggestions made by researchers and practitioners [120, 80]. Furthermore, the PD was intended to strengthen communities of practitioners: teachers with more experience in teaching the subject area would share resources and insights with colleagues to meet the constraints of local contexts [120].

The summer institute PD began by establishing a shared understanding within the community of practitioners. Elementary school teachers who had taught the CS curriculum defined computational thinking and articulated how to scaffold instructional practices for diverse learners. They noted similar linguistic needs of students new to the CS curriculum, including vocabulary development and increased exposure to the language of the discipline. Researchers and teachers then collaborated to develop linguistic frames to scaffold both the academic language related to CS concepts and the functions of social interaction. The design of language support was operationalized through the following process: 1) a researcher (first author) modeled a 15-minute CS mini-lesson on algorithms in which teachers and administrators played the roles of students; 2) a language recorder (third author) took note of the types of language use that occurred during the lesson; 3) teachers were explicitly asked to focus on language usage; 4) researchers guided the teachers in discussing actual language use versus desired language use; and 5) the group worked together to develop language scaffolds for the mini-lesson. During the post mini-lesson discussion, the teachers noticed that they used their everyday language to discuss the concept of algorithms. While students initially use their everyday sense-making abilities to access CS concepts, explicit teaching of the corresponding

language forms and functions was integrated to reinforce student understanding and develop their linguistic repertoires. Although we provide explicit teaching of language functions, we go beyond the mere use of linguistic scaffolding to conceptualize a translanguaging stance that builds on students' everyday language practices and existing sense-making repertoires [199]. To this end, we use inductive approaches to learning that engage students in sense-making through multiple modalities, which has been shown to promote multilingual student understanding of computing concepts [85]. These practices when used in concert with one another are empirically supported and effective for engaging multilingual students in STEM [27].

After this PD activity, we worked with teachers to develop language scaffolds for each lesson of the five-unit curriculum and aligned the sentence frames with the ELA and ELD standards. These scaffolds were initially designed for use in peer feedback, articulating driver and navigator roles during pair programming, and asking for assistance during the debugging process. Our PD also emphasized issues of equity and access, following suggestions of other CS teacher training programs [80]. During a gallery walk, teachers read and reflected on statistics of K–12 CS education, such as "93% of parents want CS education for their children, but only 40% of schools offer programming" [70]. Teachers commented on problems of practice specific to their contexts that challenged these statistics. For example, they pointed out that parents in their classes might be hesitant rather than enthusiastic about CS education because they equated computing with gaming. A discussion of how to increase parental buy-in occurred organically. For example, teachers suggested providing an opportunity for students to showcase their computational artifacts to their families and the greater community. These artifacts would link computational thinking concepts to students' experiences at home (e.g., storytelling and community-based projects) and in school (e.g., games that provide upper elementary students opportunities to teach their K–2 classmates how to calculate fractions).

2.6 Method

2.6.1 The Current Study

The University of California, Irvine partnered with a the Orange County Department of Education and the Santa Ana Unified School District to form a collaborative network of university and K–12 researchers and practitioners with the aim of promoting computational thinking for students in grades 3–5. This network functioned through principles of Design-Based Implementation Research (DBIR), designing interventions to implement, study, and refine alongside the county and district. The study was situated in SAUSD, a district among those with the highest percentages of low-income students (91%), Latinx students (96%), and students designated as English learners (63% in elementary grades) in the nation.

Context

This study took place in seven upper elementary (grades 3-5) classrooms across a large urban school district. Student demographics at the classroom level broadly mirrored those at the district level.

Participants

A total of seven teachers and their classrooms were selected in the partnership program based on their prior experience and interest in teaching CS to upper elementary students. Three of the teachers were Asian, two were Latinx, and two were White. Four taught mainstream general education classrooms; one taught a full-inclusion special education classroom serving both general and special education students with mild/moderate levels of disability; one taught a Gifted and Talented Education (GATE) class in which most of the students

had been identified as gifted, typically through channels including teacher recommendations, writing scores, grades, and achievement test scores. In all of the aforementioned classrooms, instruction was primarily in English, but students were consistently encouraged to leverage their linguistic and semiotic resources during CS learning. The seventh teacher taught a dual-immersion classroom composed of students who predominantly spoke Spanish at home, and many of whom were designated as having mild to moderate disabilities. She was a Mild/Moderate Special Education teacher who provided multiple opportunities for instructional and socioemotional support to meet the needs of her students. She was Latina and bilingual in English and Spanish, and she provided bilingual instruction to her students during the piloting of the curriculum. Participating teachers had extensive experience teaching multilingual students and were provided regular districtwide training on serving this population. All the students in these seven teachers' classes (total N = 108) participated in the project and thus were part of the study.

Implementation

The participating teachers piloted the year-long, five-unit computational thinking curriculum in their classrooms once a week for a lesson duration of fifty minutes.

2.6.2 Data Sources

Research Design

We utilize a mixed-methods sequential explanatory design in which the quantitative phase of data collection is followed by the qualitative phase [92]. The rationale of the sequential explanatory design is that it provides better understanding of the research problem, as the qualitative data can be used to clarify and explain the quantitative analysis [92]. In our study,

the pre- and post-test was administered at the beginning of the year to better understand how students develop their CS identities through their participation in the yearlong curriculum. We followed the post-test with semi-structured interviews to probe more deeply into student responses and provide plausible explanations for trends observed in the survey responses.

Quantitative Data

Items from the validated survey Is Science Me? (ISM) [76] were adapted to capture students' attitudes towards CS disciplines and careers and the influence of families and peers on student identification with computing (the adapted survey was renamed Is Computer Science Me? [ICSM]). The constructs were grounded in research on the roles of family support [76], school experiences [145], and self-perceptions [65]. The survey was presented in English using informal language appropriate for students with strong informal spoken and written language proficiency. Most survey item responses consisted of 3-point Likert scale items, with validating factor analysis to establish moderate to high levels of internal consistency via Cronbach's alphas and McDonald's omegas [72]. Categorical aggregates were calculated by summing responses across items relating to (1) Experiences with Computers, (2) Perceptions of Computer Science, (3) Self-Perception as Computer Scientist, (4) Family Support for Computer Science, and (5) Friend Support for Computer Science.

Qualitative Data

To develop the interview questions, we constructed open-ended questions based on the constructs underlying the ICSM survey to gain a more in-depth understanding of student responses. We paid special attention to how interview findings might relate to the quantitative survey findings. For the interviews, we selected from each classroom four students designated as English learners (N = 18), two with bridging programming experience and linguistic profi-

ciency as indicated by the California English Language Development standards, and two with emerging programming experience and linguistic proficiency. We relied on teacher judgement to select students and encouraged teachers to use the California English Language Development standards in selecting students. Teachers had intimate knowledge of these standards as they were used to develop the linguistic scaffolding embedded in the instructional materials.

2.6.3 Data Analysis

Research question one was addressed by contrasting pre-test and post-test responses to the ICSM survey. The mean post-test minus pre-test difference in student response, its standard error, t-Statistic, and effect size was calculated for each individual survey item, as well as for the categorical aggregates. Significance of the mean-difference was evaluated using a t-test, since the sample size is sufficient to justify asymptotic approximations. Unreported results using Wilcoxon signed-rank tests yielded identical findings in terms of the significance of differences across items and aggregates.

Research question two was dedicated to data analysis using top-down and bottom-up qualitative coding [172], starting with categories from prior research on our theoretical framework for developing CS identities in multilingual students and refined based on emergent themes from the study. Two researchers collaborated during the first cycle of coding to assign preliminary codes to text that pertained to theory on student identification with CS. Upon coding five interviews, the two researchers then convened to discuss and consolidate the preliminary codes. After this discussion, the lead author applied the consolidated codes to the remaining interviews, generating new codes when text pertaining to the research questions did not match the existing codes. During the second cycle of coding, the researchers combined codes into categories and subcategories to reveal emerging themes of the study. After coding all of the interviews, two researchers (first and second author) selected 10%

of the interviews and conducted an interrater reliability check. Upon initial coding of the interviews, the two researchers reached 91% agreement.

2.7 Results

2.7.1 Breakdown of Student Demographics

To better understand students' previous experience with CS, we provide a breakdown of student demographics based on their access to computers, the education of their parents and siblings, and their parents' and siblings' exposure to CS careers (Table 2.1).

2.7.2 ICSM? Survey Results

Figure 2.1 presents students' average pre- and post-survey responses for the categorical aggregate items in the ICSM survey, along with their 95% Confidence Intervals. Inferential statistics indicate positive growth in students' perceptions of their computer science identities. The categorical aggregates showed significant differences from pre- to post-survey favoring students' experiences with computer science ($Mdiff = 0.33$, $t(107) = 2.75$, $p < .01$), their perceptions of computer science ($Mdiff = 0.45$, $t(107) = 2.50$, $p = 0.01$), their per-

Table 2.1: Student Demographics

Variable ($n = 103$)	Mean
Have Computer Access	84%
Mother Attended College	52%
Father Attended College	40%
Sibling Attended College	28%
Mother has Computer Science Career	10%
Father has Computer Science Career	10%
Sibling has Computer Science Career	11%

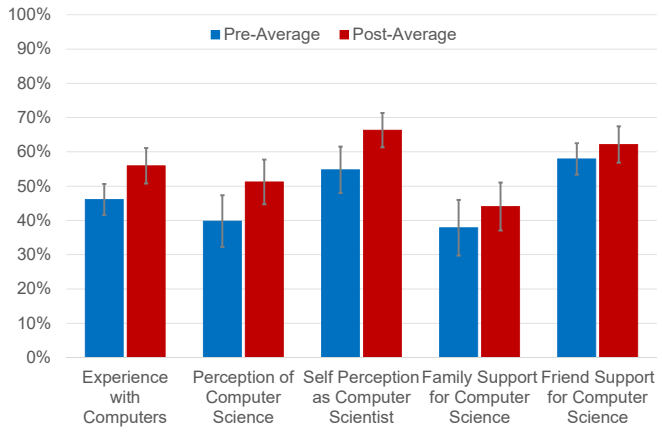

Figure 2.1: Aggregate Pre- and Post-Survey Average Responses

ceptions of themselves as computer scientists ($Mdiff = 0.42$, $t(107) = 2.39$, $p = 0.01$), and family support for computer science ($Mdiff = 0.61$, $t(107) = 3.13$, $p < .01$). Students also reported more support for computer science from their friends, but these results were not significant ($Mdiff = 0.32$, $t(107) = 1.27$, $p = 0.10$) (see Tables 2.2 and 2.3).

Table 2.2 reports average post- minus pre- survey differences in response along with the effect size and t-Test results for the individual items in the first three categorical aggregates relating to student experience with computers, perceptions of computer science, and self-perceptions as computer scientists.

In terms of student experiences with computers, the item that showed the most substantial change related to the frequency with which students talked with friends and family about computer science ($Mdiff = 0.25$, $t(107) = 3.91$, $p < 0.01$). Two other items that showed insignificant improvements were whether students take apart toys and computers to see how they work and whether they use tools to build things. The item asking students whether they write computer programs showed a negligible decline.

In characterizing students' perceptions of computer science, there was a significant increase from pre- to post-test in the degree to which they believe computer scientists are respected ($Mdiff = 0.18$, $t(107) = 2.63$, $p < 0.01$) and make a difference in the world ($Mdiff = 0.23$,

Table 2.2: Student Interest in Computer Science

Panel A: Experiences with Computers					
$n = 108$	Mean Diff	Effect Size	Std Error	t-Statistic	p-value
I write Computer Programs	-0.01	-0.01	0.06	-0.16	0.56
I talk with friends and family about CS	0.24	0.38	0.06	3.91***	<0.01
I take apart toys, computers to see how they work	0.03	0.04	0.07	0.42	0.34
I use tools to build things	0.07	0.12	0.06	1.27	0.10
Sum Categorical Aggregate	0.33	0.26	0.12	2.75***	<0.01

Panel B: Perceptions of Computer Science					
$n = 108$	Mean Diff	Effect Size	Std Error	t-Statistic	p-value
I am good at CS	-0.02	-0.02	0.07	-0.26	0.60
I think CS is interesting	0.06	0.09	0.07	0.90	0.19
Computer scientists are respected	0.18	0.25	0.07	2.63***	<0.01
Computer scientists make a difference in the world	0.23	0.29	0.08	2.99***	<0.01
Sum Categorical Aggregate	0.45	0.24	0.18	2.50***	0.01

Panel C: Self Perception as Computer Scientist					
$n = 108$	Mean Diff	Effect Size	Std Error	t-Statistic	p-value
I can learn CS	0.12	0.18	0.07	1.84**	0.03
I don't like to do things I can't master quickly	0.09	0.09	0.10	0.91	0.18
If people tell me I can't do something, I try harder	0.12	0.13	0.09	1.38*	0.08
I enjoy trying to understand difficult things	0.08	0.10	0.08	1.03	0.15
Sum Categorical Aggregate	0.42	0.23	0.17	2.39***	0.01

$t(107) = 2.99$, $p < 0.01$). The survey also registered negligible changes in students' belief that they are good at computer science and think computer science is interesting.

Students' self perceptions as computer scientists showed significant increases in the aggregate ($Mdiff = 0.42$, $t(107) = 2.39$, $p = 0.01$) due to moderate increases in all items. Students demonstrated a statistically significant increase in their belief that they could do computer science ($Mdiff = 0.12$, $t(107) = 1.84$, $p = 0.03$), and statistically insignificant increases in their willingness to do things they can't master quickly ($Mdiff = 0.09$, $t(107) = 0.91$, $p = 0.18$), willingness to try hard in the face of adverse signals ($Mdiff = 0.12$, $t(107) = 1.38$, $p = 0.08$), and enjoyment of understanding difficult things ($Mdiff = 0.08$, $t(107) = 1.03$, $p = 0.15$).

Table 2.3 reports average post- minus pre- survey differences in response along with the effect size and t-Test results for the individual items in the last two categorical aggregates relating to family and friend support for computer science.

Students' perceptions of family support for computer science increases substantially ($Mdiff = 0.61$, $t(107) = 3.13$, $p < 0.01$). Comparing post- with pre- survey responses demonstrate significant increases in the importance students' families assigned to getting good grades ($Mdiff = 0.13, t(107) = 2.32, p = 0.01$) and trying their best ($Mdiff = 0.12, t(107) = 2.94$, $p < 0.01$), as well as their family's knowledge of their school performance ($Mdiff = 0.22$, $t(107) = 3.50$, $p < 0.01$). These responses also showed insignificant increases in the extent to which students' families thought computer science was important for them to learn ($Mdiff = 0.10$, $t(107) = 1.49$, $p = 0.07$) and interesting ($Mdiff = 0.04$, $t(107) = 0.51$, $p = 0.30$).

In evaluating students' friends support for computer science, the categorical increase was not statistically significant ($Mdiff = 0.32$, $t(107) = 1.27$, $p = 0.10$). Students did show statistically significant increases in the extent to which their friends thought computer science

Table 2.3: Community Support for Computer Science

Panel A: Family Support for Computer Science					
$n = 108$	Mean Diff	Effect Size	Std Error	t-Statistic	p-value
My family thinks CS is important for me to learn	0.10	0.14	0.07	1.49*	0.07
It's important to my family that I get good grades	0.13	0.22	0.06	2.32***	0.01
It's important to my family that I try best	0.12	0.28	0.04	2.94***	<0.01
My family knows how well I'm doing in school	0.22	0.34	0.06	3.50***	<0.01
My family thinks CS is interesting	0.04	0.05	0.07	0.51	0.30
Sum Categorical Aggregate	0.61	0.30	0.20	3.13***	<0.01

Panel B: Friend Support for Computer Science					
$n = 108$	Mean Diff	Effect Size	Std Error	t-Statistic	p-value
My friends think CS is cool	0.21	0.19	0.11	2.01**	0.02
My friends encourage me to do well in school	-0.16	-0.11	0.13	-1.17	0.88
My friends like computer science	0.27	0.23	0.11	2.41***	0.01
Sum Categorical Aggregate	0.32	0.12	-0.26	1.27	0.10

is cool ($Mdiff = 0.21$, $t(107) = 2.01$, $p = 0.02$) and how much their friends like computer science ($Mdiff = 0.27$, $t(107) = 2.41$, $p = 0.01$). However, there was a decline in the extent to which students' friends encourge them to do well in school ($Mdiff = -0.16$, $t(107) = -1.17$, $p = 0.88$).

2.7.3 Results from Student CS Identity Interviews

Students' experience with computers. Kang et al. [102] underscore the influence of students' personal and family backgrounds on disciplinary identification. Family factors, especially parental support, represents a key factor contributing to disciplinary identification [8, 212, 10, 141, 146, 183, 184] The survey item with one of the largest increases was "I talk with friends and family about CS." These discussions provided multiple opportunities for children to perceive their families showing their support for CS and becoming more involved in their childrens' learning. Three major themes emerged from the analysis of parental

involvement, which were rooted in their positive perceptions towards CS, strong interests in CS, and viewing and creating work with their children.

An examination of how students believe their parents think about computer science revealed the level of importance they assigned to learning of discipline. For example, one respondent said:

> "My Mom's like, it's great because since you're already learning it, then you get the hang of it and when you're an adult you already have the hang of it. You are going to be like, "Oh, I already learned this in second grade, and then if I learned it in fifth grade, then I will learn [more] each year!"

This excerpt suggests that the student perceived their parents to be supportive of their CS learning and viewed it as valuable preparation for future learning. Students also reported showcasing their projects to their families, who expressed their appreciation of the computer science curricula in its ability to realize their children's creativity and ideas.

> "My mom was impressed...She was impressed because I went, "Mom, look! I learned something new in class!" She's like, "Oh that's really impressive!" And then she's like, "Oh It must be fun!" And I'm like, "Yeah, it's pretty fun." She's like, Oh show me the um, the projects I had been doing. They're like, oh, really cool. What you did. Like they like everything I do and they're like, really? They, they call me like really creative."

"

As students like the respondent above experience computer science across multiple contexts (at school, at home, with family), they begin to develop positive perceptions about how they themselves fit within CS communities [53, 139]. In addition to positive parental perceptions

of CS, students discussed the benefits of talking about CS with their parents at home. For example, over half of those interviewed reported working together with their parents on Scratch projects.

> "I talked to my mom and my dad. How fun it is sometimes if we can do projects together."

A small number of students said they had spent extended lengths of time working on complex projects with their parents.

> " I kind of do a lot of outside programming with my, sometimes me and my dad do like games on scratch...We do like really fun games. Like we tried making our own Fortnite game but it came out really funny. . . .It's really funny the look of...how I edited it because I edited it. . . .It's kind of like a Pac Man. Characters are like Pac Man and then I got the idea from him because of his game...Yeah he helps me code a lot. He helps me with the levels...He's not a computer scientist, but he's fun"

A reasonable explanation for increased student perception of parental support could be the association they draw between classroom learning and the values and skills present in their families [193]. Students may practice computer science with family members who also have interest in games, providing entry into computer science not only for students, but also for families who might not otherwise have experience with the discipline.

Some students also reported practicing computer science with their siblings and extended family and friends.

> "I do it at my house. I did it. I, um, my grampa's friend's house, I did it basically

everywhere I can take my Chromebook. I don't really get to use electronics so I just like being able to use Scratch."

"Yeah like...we did one [Scratch project] that was called "how I live with my brother and, and, and how I live with my sister". It was fun. We did it together. He said it's fun living with my sister. I liked it a lot because he's like creative."

"My cousins, they showed me Code.org and I showed them Scratch. I show my cousins and sometimes my mom lets my friends come into my house? Yeah. Like in my friend Tina. Sometimes she visits and we don't know what else to do so we do Scratch."

Again, these students practiced CS learning beyond the traditional classroom lesson or after-school coding club. As students began to develop their CS identities, their experiences with immediate and extended family helped to solidify their disciplinary identification. Together, these results provide important insights into how students' parental and family involvement influenced their experiences with computer science and how those experiences shaped their identification with the CS discipline.

Students' perceptions of computer science. Early exposure to computer science facilitates students' perceptions of what it means to be a computer scientist and to do computer science [53, 139]. The reciprocal relation between students' perceived selves and imagined future selves in the field of computer science [102] are often governed by students' perceptions of what it means to be a competent CS actor [34]. From the interviews, we saw that the majority of students had an accurate understanding of who computer scientists are and what they do. When the participants were asked what computer science is, over half of them made connections between computer science and programming or coding. Furthermore, they provided multiple examples of what computer scientists do, such as studying

computers, creating games, making apps, developing websites, and building things. A small number of those interviewed indicated that their own experiences with coding acted as fulcrums for further exploration. For example, one student imagined computer scientists as "making programs for kids to learn," which enabled her to identify with the profession in an age appropriate manner.

> "They start making programs for kids to learn. And then they make programs...so like kids can see like what they do and then they could have like a background of like what they do."

This student believes that the child-friendly presentation of computer science content has provided her with foundational knowledge of the types of practices adult computer scientists participate in. The age-appropriate curricular content and Scratch programming language provided a bridge to identification with a profession traditionally viewed as for adults only.

Computer scientists make a difference in the world. Students' belief that computer scientists make a difference in the world demonstrates how sociocultural aspects of computing go beyond classroom dynamics to encompass broader social issues. Students presented a variety of reasons why they believe that computer scientists make a difference in the world, including providing career opportunities, normalizing the making of mistakes, ensuring ethics and safety in computing, innovation, and making production more efficient.

Many of the students provided economic reasons for learning computing.

> "They can inspire people in that they teach about coding and about other jobs that require coding so they could learn more about it."

> "I think people that do coding, it could help people understand more about it and...really get them to like...learn more about it, for like one day they could do

it too and like probably get a better job."

Still other students believed that the normalization of mistakes was a valuable approach to learning that not only garnered respect, but actually made a difference in the lives of others.

> "If you do mistakes you can change them...So that really makes a difference, that you can change your mistakes."

> "So like kids can just like have fun doing stuff. If they make a mistake they don't have to worry, they can just go back and redo it."

One respondent mentioned the need to make the internet a safer place. When asked if they think computer scientists make a difference in the world, they said:

> "I'm hoping the internet and making the internet safer...maybe they'll stop like cyber bullying, how they could like, uh, I don't know, but maybe they could stop people from like being mean to other people on the Internet and report them to the, to the owners of like, like if anybody is being really mean to anybody on Facebook, they can report it to the owners and the Facebook app so that way...nobody will have like to feel bad."

Finally, one student was able to see how CS concepts are at work in our everyday lives.

> "Um, like making things by computer science, like coding things so we don't have to repeat things....Like if you're building a wheel for a car and there's four wheels. Like, um, if you had a blueprint instead of having to write it four times, if he just like coded it out instead of having to repeat it four times."

This student's understanding that CS concepts are present outside of programming environments allowed them to explain how concepts such as loops could help society by making production more efficient.

Student interest in computer science. While the survey showed no significant difference from the pre- to post-test item measuring interest in CS, in the interviews students shared a variety of explanations as to why they thought CS was interesting, including their ability to express themselves freely, and the opportunity to leverage their varied resources to learn CS.

Many of those interviewed reported strong interest due to the freedom of expression embedded into instructional activities. The relatively unrestricted levels of choice embedded in the Scratch interface seemed to create positive perceptions of computer science and foster student ownership of their projects. This was made possible by the opportunities to make innovative design choices, and to take an iterative approach without facing negative feedback or criticism.

> "I like [computer science]! It's fun to do when you're like bored sometimes cause you can create and then do what you want...Um, you can like erase it and you can do like anything...you can control. Yeah. Yeah. You can make your own thing right. And then you can change the name on it...Like my stuff. The sprites that you want."

This theme came up in discussions of how students personalized their projects. For example, one interviewee described how she created her own game as part of a Code.org challenge (Flappy Bird) that was embedded in the curriculum.

> "So, um, in the Flappy Bird, um, game that we did, we can choose different types of sprites to do it. We did different types of code. So, for example, I did "If hit

> the ground and you would lose the game and then you restart and then, um, the wallpaper would change when you made a score."

By adding personal touches to her programs, and personalizing her work, this student was able to leverage her creativity and imagination to grasp complex computational concepts, such as conditional logic. The comment below illustrates why many students reported liking CS: they can personalize it and be creative in their work.

> "I like doing computer science because you get to create something on your own and you can personalize it and try to, you can try to be creative with doing it like that."

These students' creativity, however, did not reflect exclusively internal mental processes. On the contrary, their creative activities were social in nature as many students reported multiple sources of inspiration for creating their own projects.

> "I have access to a computer, so sometimes I go scratch and explore others. I look at other projects and use them and see how they created it."

Taken together, these results suggest that students were motivated to take ownership of their projects, and were able to find inspiration from both their own ideas and from reusing and remixing the work of others.

A final recurrent theme relating to students' interests in CS was a sense among the interviewees that the curriculum built upon their prior knowledge and experiences. The most commonly reported explanations of why CS was fun related to leveraging their interests, such as their gaming experiences, imagination, proclivities for building and making, creativity, expression of ideas, and desire to mentor others. One participant said:

> "It's fun coding things because you can imagine. Your imagination comes to you when you're coding everything and then you code everything that you always imagine."

This view was echoed by another participant, who identified the different modes of expression afforded by Scratch (i.e., stories, conversations and games) as representing a key motivator. As this interviewee put it:

> "I try to make stories, conversations, and try to make games that include what I like. I already made a game, it is like a unicorn going to different backgrounds, and it makes music...and I included that because I like unicorns. It's like a way to express like what I [want], like when not like telling it in words but showing it."

In this excerpt, the interviewee prefers to express herself by showing not telling; she leverages her own semiotic resources to facilitate self-expression. This content-first approach to learning encourages students to access the discipline before engaging in unnecessary linguistic tasks, which may avert resources away from learning. There is a growing body of asset-based research on multilingual students illustrating how they draw on a range of resources (i.e., linguistic, cultural, semiotic, embodied, etc.) during CS learning activities [199]. Instructional practices that leverage these resources draw on multilingual students' full repertoires for learning, sense making, and identity development [199], thereby increasing their opportunities for full participation in CS. Together, these results provide important insights into how leveraging multilingual students' resources shapes how they envision themselves in CS.

Students' perceptions of themselves as computer scientists There was a sense of self-efficacy among interviewees with regard to their perceptions of themselves as computer scientists. Survey results indicated a significant increase in students' beliefs that they can

learn CS. When asked whether they can learn CS, the majority of participants responded in the affirmative and expressed a variety of perspectives as to how this learning was beneficial for them.

For example, one interviewee said:

> "Cause if sometime in the future I want to be a computer scientist, I would already have an experience with the computer. So it would be kind of easier for me to like, you know, like to do stuff and it wouldn't be anything new. Yeah, cause one day, if I ever grow up, I want to be like one of those guys that makes games."

Again, we see how students leverage their own experiences (e.g., making games) to make connections with the CS profession. There were some suggestions that learning CS would enable them to teach their peers. Another interviewee, when asked whether they could learn CS, said:

> "I would help people learn, like about coding out and tell them about the event blocks and about the characters and about how to change backgrounds and how to take care of the internet and put it [the Scratch project] on there."

In sharing his expertise to support his classmates, this student developed a connection between his perception of self and the discipline of CS. The social recognition that accompanies identities marked by developing expertise provides further learning opportunities that may not arise without this type of recognition [19].

How positive attitudes toward making mistakes fostered persistence One of the biggest affordances of the curriculum was teaching students to persevere when they made mistakes. In the survey, all students showed an increased belief in their ability to persist

in the face of difficulty, and the item "If people tell me I can't do something, I try even harder" showed significantly positive growth. Students reported a variety of reasons for persisting in debugging their code by participating in the following activities: asking for help, applying multiple strategies including trying new blocks, and applying complex problem solving strategies.

For some of these students, help seeking was the strategy used to resolve initial feelings of frustration. For example, one student stumbled across a bug in which she needed a negative 10 as opposed to a positive 10. She reached to her teacher for help and learned about negative values. She expressed shock that a program can experience so many problems because of one little line (the negative sign).

> "I think...when it was supposed to be negative 10 but I put 10 and not the negative one. And then when [my teacher] looked at her cheat code, it was negative 10 and then I think just for that little line, I couldn't do it?! Yeah. It's funny how it's one thing, but it's making a big difference in your code."

Although Scratch is a block-based program, it can still provide syntactical challenges for students. When asked what types of strategies she used when encountering problems in general, this participant reported trying new blocks or generalizing the advice her teacher gave her by switching positive and negative values in her code. This suggests that students may tend to overgeneralize strategies learned, indicating a greater need for strategy training in debugging exercises. Nevertheless, a variety of themes emerged from the analysis of student strategy use when encountering mistakes. The strategies students typically employed involved differing degrees of complexity, from trying new blocks and asking for help to experimenting and iterating on projects and engaging in planning and abstraction [25]. The most commonly reported technique reported for debugging mistakes was trying new blocks.

> "If it didn't work I would've tried it again, but like with a different block...different, let's take one of them out. Which one didn't seem to [work] and try different blocks."

Turning now to more complex problem solving techniques, a small number of participants used more advanced computational thinking practices to debug programs. In one case, a student participant persevered after making mistakes by applying multiple metacognitive strategies. When she encountered a mistake, she viewed it as a way to practice abstracting and modularizing [25], that is, she would decide which parts of the project she wanted to keep and which to exclude, make a plan, and then tinker and iterate with it to makes changes depending on her design preferences.

> "Sometimes I do [make] mistakes, but um, I always make sure that if I did make a mistake, I come back to it and try to fix it and solve the problem. Um, I try to see...what I want, exactly what I want, I try to make a plan of what I want. Then I kind of played with it and see if I liked it and then I'll change stuff...if I liked something else"

In the above example, she referred to practicing identifying the problem within her code and on the user end to identify the problem while contextualizing these decisions based on her preferences.

Overall, students had a positive view of mistakes, most often viewing them as chances to learn something new.

> "Like some people say that mistakes are our friends, they help us and then we can figure it out. Like, and learn something...So mistakes are like good for us. They help us a lot... like if you make a mistake in math or science, [you] can learn from that and try over again and get a better answer."

There were a few participants who reported overcoming the phobia associated with programming and working with computers in general. When asked about how they felt making mistakes, one participant said:

> "I feel like it's all right because it's not like going to harm me. I feel like it's just that sometimes I need a little bit more practice. "

Finally, some students even leveraged their mistakes as an opportunity to improve their programs.

> "When I made a mistake, I tried to fix it...in a way that helps me...like get it better. Like so then, so then I won't have to like keep...the same idea."

In summary, for the participants in this study, mistakes provided more opportunities than pitfalls. Students were able to overcome their initial frustration to apply multiple strategies to debugging their code. Furthermore, they changed their orientation toward mistakes from negative to more positive views, seeing them, for example, as opportunities to learn and improve their work.

Support from peers. When the participants were asked what their friends thought about CS, the majority commented that their friends think CS is fun and exciting. Survey results corroborated this finding, as the item "My friends like computer science" showed a statistically significant increase. Furthermore, students expressed a variety of perspectives regarding how their friends provided support when they shared their work. Many students discussed the feedback they received from friends when showcasing projects. For example, one respondent said:.

> "Since they say it's good, they gave me suggestions too. Um, so like I, before I

had the [sprite] move in my About Me project, I told them about it and they said, why don't you make something move?"

In addition to providing feedback, interviewees also reported receiving compliments from their friends on their work.

"The first day when I told them about the About Me project, they're like "Oh, I'm pretty impressed about this!"

Comparing the two results, it is clear that peer support created a positive influence on student learning in relation to building a sense of efficacy and community around their work.

Out-of-school learning environments. The final part of the interview included questions about students' out-of-school coding experiences. In many cases, they discussed practicing CS outside the classroom, such as at after-school coding clubs. One student reported working on her own time, *"one day yes, one day no, when I have time."* Furthermore, she drew inspiration from remixing and reusing ideas from her classmates. *"I always have an idea. I always look at my classmates and then I see what they're making...and they give me an idea and then I do it my own way."*

In another example, a participant made a racing game in their teacher's class and discussed how proud they were of it.

Student: "It's fun coding things because you can imagine...your imagination comes to you when you're coding everything and then you code everything that you always imagine. Then you can feel very proud when you make a game."

Interviewer: "That's awesome. Which one has been your favorite project so far or the one that you feel most proud of?"

Student: "Um, it was a racing game that I made."

Interviewer: "Oh, nice."

Student: "Um, Mrs. Jeanie's Coding Club afterschool."

Interviewer: "What did you do? That one there?"

Student: "Yeah."

Interviewer: "Nice...What are the rules for your racing game? How did you make it go?"

Student: "One person can use the arrow keys, the up, down, left, and right arrow keys and the other one has used w, a, s, and d...and You had like a track and then whoever makes it first wins."

Students are typically proud of their work when they are able to extend their learning beyond the curriculum, while the extra practice helps to hone their skills and build confidence. Another student alluded to practicing Scratch at home, providing multiple reasons for going beyond what was assigned to create additional projects.

"I would use it for like, when I'm bored, when it's in an assignment, when I want to, when it's fun"

Comparing these two results, it can be seen that students practice computer science outside the classroom for multiple reasons, including using their imagination, alleviating boredom, practicing their hobbies, and having fun. Interestingly, some students extend their learning beyond the classroom by seeing CS concepts in everyday life.

"Um, like if you're like walking, you don't just take one step, you take multiple steps so it's like a loop."

This is a rather remarkable outcome as the student is able to contemplate computational thinking in the absence of explicit programming or curricular activities.

Finally, some students advanced their understanding by searching on the internet for CS-related content, such as science and engineering videos.

> "I go to coding videos and watch science and engineering...I watch flocabulary... videos on, do you know of Flocabulary? It basically has science...we do it in our class."

Overall, these results indicate that CS learning extends beyond traditional learning environments to encompass how multilingual students' learning and identity development is shaped by participation outside of school, at home, after school, and within their communities.

2.8 Discussion

The findings from this chapter are corroborated by previous literature highlighting the role of family support [76], self-perception [65], and formal and informal learning experiences [145] in developing student identification in STEM. These results also build on previous research that demonstrates the importance of early intervention in developing students' later interest in CS careers [81, 191]. Finally, the findings point to the strong need to engage multilingual students in disciplinary practices that leverage their existing resources, which reduces the burden of having to learn language and CS at the same time [117, 122, 121]. By integrating practices shown to be effective [27], we provided a context in which existing resources could be used to make CS learning more immediately relevant and valuable to students. This has resulted in students pursuing CS learning across a variety of formal and informal educational settings.

2.8.1 Students' Experiences with Computers

Findings from this chapter suggest that multilingual students' discussions of CS with family and friends helped to normalize it and contributed to greater identification with the field [97]. While family and community engagement are critical to providing responsive CS education [8, 212, 10, 141, 146, 183, 184], the current focus on testing, which constrains other disciplines such as math and science, leaves little time for CS learning and fails to account for the sociocultural processes that underlie multilingual student identity development. Sharing projects with family and friends has had the reciprocal advantage of shaping how students perceive being seen by members of their households and communities, namely, as having CS expertise. Furthermore, the time students spent sharing projects and talking about CS outside of school provided more opportunities for them to perceive themselves as capable and invested participants in CS communities. As CS begins to be systematically implemented in elementary grades, it is incumbent upon educational policy makers, curriculum developers, and practitioners to highlight the role of family and peer support in developing students' interests.

Furthermore, the findings revealed that students began to associate classroom learning with the values and skills present in their families. These findings are consistent with recent research in computing, maker spaces, and other scientific disciplines [154, 90, 193]. When students find their work to be personally meaningful, they begin to make connections between their own knowledge and experience and the curriculum. In a similar study, indigenous boys made connections between computational principles and working with their parents on car mechanics [175]. Findings such as these highlight that, to encourage students' engagement with CS, it is crucial to provide several points of connection that draw from students' rich cultural and familial traditions and wealth of knowledge. These findings are corroborated by prior research into the role of familismo [163], meaning strong connections to immediate and extended family that value collaboration and community, in positively influencing the

development of STEM identities for predominantly Latinx students [163].

2.8.2 Students' Perceptions of Computer Scientists

Multilingual student identification with the discipline is contingent upon their perceptions of who competent actors are and how they value these actors' roles [53]. Students had an accurate understanding of who computer scientists are and what they do, and they described numerous ways in which computer scientists make a difference in the world. In perceiving how computer scientists contribute to solving broader social and economic problems, they developed a better understanding of how computing can be used to solve problems in the broader community. As a next step, curriculum developers could leverage these perceptions to encourage students to use computational thinking to solve problems within their own communities. Through efforts such as these, students have the potential to transform computational thinking into computational action that has direct impact in their own communities [192]. Furthermore, as part of the pre- and post-survey, we asked students the following open-ended questions: "Who are computer scientists and what do they do?" We plan on analyzing these data for future research to understand how students' understanding of computer scientists grew both as a class and at the individual-student level.

2.8.3 Students' Interest in Computer Science

While the survey results indicated that students did not significantly increase their interest in CS, the interviews painted a different picture. Findings from the interview are consistent with previous findings that self-expression [93, 25, 154] and the ability to leverage their varied resources [117, 122, 121] are key contributors to developing student interest in computing. The personal choice integrated into Scratch-based activities and projects played an integral role in disrupting stereotypes about who does CS [101]. As students personalized their work,

they began to take ownership and envision themselves as producers of computational artifacts. Furthermore, students' ability to showcase their knowledge instead of "telling" it was reflected in the curriculum's content-first approach, which provides equitable points of access for multilingual students to engage in complex content [118]. An underlying contribution of this content-first approach to learning is that students are encouraged to access the discipline before engaging in linguistic tasks, which may unnecessarily avert resources away from learning [118]. It is through content learning that students interact and communicate with one another, beginning to co-construct knowledge, and thus become active creators of disciplinary practices. To this end, the principles underlying the practices embedded in our curriculum include examining what STEM subject matter "does," not as a codified body of knowledge, but as a vehicle for making sense of complex problems and phenomena [215].

Furthermore, previous research on multilingual student engagement in STEM has underscored the importance of engaging these students in meaningful interaction that provides authentic contexts for language use [27]. In our study, student identity construction was social in nature: as their project design and development did not occur in isolation, but instead was the dynamic result of concerted efforts from teacher and peers. This finding is corroborated by theoretical works and empirical studies that take a sociocultural approach to identity development [34, 137, 97]; as students view their learning as collaborative and relational, their disciplinary interest strengthens.

2.8.4 Students' Perceptions of Themselves as Computer Scientists

Students developed a greater sense of efficacy in their ability to learn CS, rooted in making connections to the profession and positioning themselves as experts within the field. Such findings are consistent with previous research that highlights how positioning students as experts not only increases their efficacy beliefs [34] but also provides further opportunities for

learning. Dorner et al. [61] examine how predominately Latinx, multilingual students often act as language brokers at multiple socialization sites, such as the home, in which immigrant parents may not speak English, and the classroom, which presents a site for dominant use of the target language. These brokering opportunities provide students opportunities to position themselves as experts within these sites and foster stronger disciplinary identities [69, 158]. As this study represents the exploratory phase of a larger effort to refine, test, and scale the computational thinking curriculum, we have added additional content to the curriculum to encourage students to see experts from similar cultural and social backgrounds as themselves, enabling students to see themselves within the field of CS. One revision to the curriculum has been to add "Memorable Mentor" videos, in which students learn about programming from predominantly Latinx computer scientists, and then reflect on how these experts' work is relevant to solving problems globally or within their local communities.

2.8.5 Students' Perceptions of Making Mistakes

Unlike math and science, CS frequently proposes multiple solutions to a single problem, which results in reframing "mistakes" as potential learning opportunities. This has implications for equity, as multilingual students may use there everyday sense making abilities, instead of discipline-specific language, to articulate their problem-solving approaches in the target language, resulting in their teachers viewing their approaches to problem solving as mistakes rather than as novel, complex, and innovative solutions [214]. Furthermore, as debugging is a recognized computational thinking practice [25], fixing mistakes in code is a principle component of CS learning. What is less well studied is how developing a positive attitude toward making mistakes fosters persistence. Recent research has shown how having elementary students embody debugging activities through unplugged activities such as walking through mazes fostered persistence in coding [6, 5]. Given the extent to which these students viewed making mistakes as opportunities to learn, further research on the

relationship between debugging and persistence in CS is warranted.

2.8.6 Support from Peers

While in aggregate, the category for peer support did not show significant growth, the item "My friends like computer science" showed a statistically significant increase. As mentioned above, disciplinary identification occurs when activities are viewed as socially typical [97]. Conversely, students tend to disidentify with a discipline when they see its activities as socially atypical. This disidentification can also spread through peer groups by way of peer pressure [187]. Due to its virulence, disidentification represents a key contributor to underrepresentation of marginalized students, such as women, Black students, and students of color [103]. It then becomes key to distinguish incidences when students "choose not to learn" from those in which they have difficulty grappling with the material, when examining the effects of disidentification on student learning [107]. Dornyei [62] finds that positive peer dynamics represent an essential component for increasing multilingual students' motivation in linguistically diverse classrooms.

2.8.7 Learning CS Outside of School

Instructional interventions such as our culturally and linguistically responsive computational thinking curriculum provide equitable CS education for multilingual students by affording a sociocultural approach that extends beyond traditional learning environments to encompass how multilingual students' learning and identity development is shaped by participation across social and cultural contexts [53, 139]. Results from the interviews highlighted how multilingual students' learning and identity development was shaped by participation outside of school, at home, after school, and within their communities. These findings are corroborated by CS and STEM education research on informal learning environments, which

presents promising opportunities for creative expression and reshaping how diverse youth view literacy, learning, and expression [154]. Research on multilingual students indicates that situating science learning within informal contexts has been an effective approach for teacher education [39, 87, 190]. Given these promising results, future research on multilingual student learning of CS in informal settings would help to uncover the factors that contribute to meaningful participation.

2.8.8 Limitations

A clear limitation of this chapter is the lack of a control group in measuring students' learning and identification with CS. This has been a key issue in elementary studies of CS learning, as CS has not yet become a credentialed subject, and opportunities for comparing an intervention to business as usual are limited. Nevertheless, the next phase of our project will compare students receiving the computational thinking curriculum to students receiving "business as usual" elementary subjects, to see that the computational thinking curriculum increased their CS achievement while having no impact on their proficiency levels in math and English Language Arts.

Another limitation is that we adapted a validated survey commonly used in science classrooms (ISM); it is possible that results are skewed in the positive direction as students already have knowledge of science before they begin a specific intervention, but may not have as much knowledge of CS. Our project team has acknowledged this limitation, and as this study represents the first phase of a larger intervention to refine, test, and scale the curriculum, we have chosen another student attitude survey to measure CS identities.

Furthermore, it is possible that students exhibited ceiling effects with respect to their interest in CS. However, the survey included several other questions, across multiple categories, that highlight how their identities grew along several dimensions. A potential strength of the

follow-up interviews is that they enabled us to discover why the students were interested in CS, despite any ceiling effects that might obscure the extent of their growth.

Another limitation of this chapter is that we did not have student- and teacher-level data due to the nascent nature of our Research Practice Partnership; therefore, we could not explore how students' backgrounds compared to those of the teachers. In the course design process of CS curricula, such as the MOOC Integrating Computational Thinking into the High School Curriculum in Puerto Rico, researchers have recruited teachers with heritage similar to students [143], which has been shown to provide greater academic and social outcomes for Latinx students [210]. Future research should focus on how teacher demographics influence student identification with the field of CS.

In addition, there are several limitations to using linguistic scaffolding when it is not implemented properly. The instructional moments in which sentence frames provide affordances include when they are used to reinforce concepts that are learned inductively. To this end, concepts should first be taught inductively in a manner that engages students in peer-to-peer interaction. During this phase of instruction, language scaffolding has been shown to stifle communication, insofar as it limits students' rhetorical choices to prescriptive language. While providing language support is integral to teaching language and CS together, there is much to learn about plurilingual approaches such as translanguaging and content-first approaches to STEM instruction that decolonize traditional views of academic language teaching by emphasizing student understanding over language usage [199, 82, 135].

Finally, as we are focusing on multilingual students, it could be argued that we should focus on their linguistic identities rather than their CS identities. We argue that the content-first approach to the curriculum and the multimodal affordances of Scratch provided entry points into the discipline that these students might not otherwise encounter in a language-heavy math and/or science curriculum. These points of access, along with the connections students made between the curriculum and the values of their families, and considering the multiple

settings in which students practiced CS (at home, with friends, at after school coding clubs, with extended family), all contributed to overall greater identification with the field.

2.8.9 Conclusions

Too often, educators assume that multilingual students come to school lacking the conceptual or linguistic resources necessary for learning CS. In contrast to these assumptions, the instructional models we embedded in the curriculum leveraged multilingual students' existing resources (conceptual, social, linguistic, cultural, semiotic, etc.) to open new possibilities for CS education. The purposefully tailored curriculum, coupled with students' home and school access to Chromebooks, provided opportunities for students to learn and identify with the materials through participation with family and peers in and outside of school. We recommend that teachers pay close attention to the dominant narratives surrounding who does CS and combat stereotypes by encouraging students to view themselves as capable participants in CS communities. This can be achieved by leveraging students' personal and family backgrounds, garnering parental support, making connections between the CS profession and broader social issues, positioning students as experts, engaging peer support networks, and providing multiple points of connection between formal and informal learning environments.

In this chapter, we demonstrated the ways in which students developed disciplinary identification with the field of CS through their engagement in a culturally and linguistically responsive curriculum. Throughout their engagement, students were able to leverage their multiple resources across formal, home, and informal learning contexts. The most encouraging part of this story is that students had multiple ways to identify with the instructional materials in a manner that leveraged their identities to support CS learning. This offers promising possibilities for educators who wish to provide early exposure to CS in a manner

that shapes student interests and inspires them to pursue the profession.

Chapter 3

Integration of Computational Thinking Into English Language Arts

3.1 Chapter Synopsis

This chapter 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, I discuss recommendations for future integrated ELA-CT curricula.

3.2 Introduction

While the integration of computational thinking (CT) into science, technology, engineering, and mathematics (STEM) education has been well studied [177, 204], there is a smaller but growing body of work on CT and literacy integration [96, 30, 99, 199]. 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) [30]. This can facilitate the kinds of inquiry, cultural and community engagement, and social recognition that are integral to fostering identity development in STEM [46]. 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 [63]. 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 chapter 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: 1. What strategies are used by upper elementary teachers to integrate CT into literacy and language instruction? 2. 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?

3.3 Theoretical Framework

3.3.1 Computational Literacies

Our study draws from Jacob and Warschauer's [96] model of computational literacy, which situates computational thinking as a fundamental literacy required for full societal participation [60, 209]. 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) [96] (see Figure 3.1).

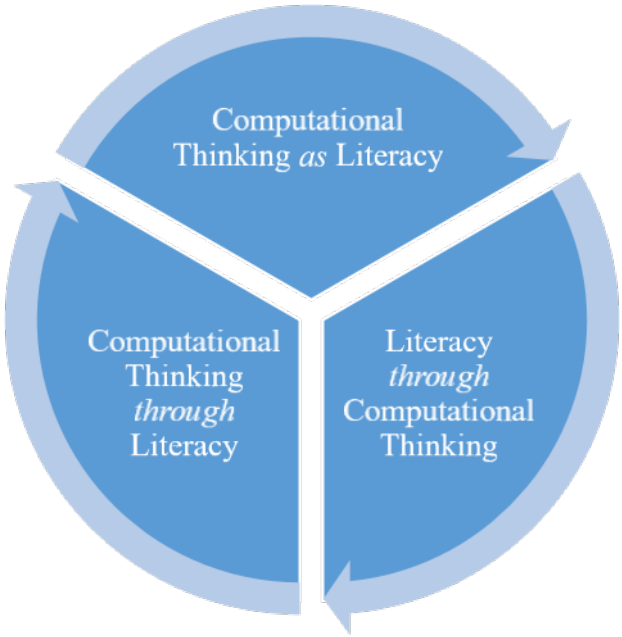

Figure 3.1: Three-Dimensional Framework for Understanding Computational Thinking and Literacy (Jacob & Warschauer, 2018)

For the purpose of this chapter, 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 [155, 20]. 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 [155].

Additionally, informational and narrative genres capture the semiotic process related to computing. To illustrate, Burke and Kafai [30] 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. [54] 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.

3.4 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 [216]. Our project has addressed this challenge by adapting the Creative Computing Curriculum [24] 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. 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.

3.4.1 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) [27]. 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 [44] while making instruction more engaging, concrete, and meaningful for multilingual students [98, 45, 166]. Computer science disciplinary activities and learning goals are aligned with standards to guide teachers (see Table 3.1 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 [179].

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 and

Table 3.1: Sample learning goals that integrate Grade 4 Common Core ELA, English Language Development, and Computer Science Teaching Association standards

Computer Science Teacher Association Standards		
Activity: Students program a story about their lives, families, or communities		
Computer Science Concepts: Loops, Sequences, Conditionals		
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 Test and debug a program or algorithm to ensure it runs as intended	
CSTA 1B-AP-15		
English Language Development (ELD) Standards		
Emerging	Expanding	Bridging
3. Offering Opinions	3. Offering Opinions	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.	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.	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	11. Supporting 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.	Offer opinions and provide good reasons and some textual evidence or relevant background knowledge (e.g., paraphrased examples from text or knowledge of content).	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	Engage effectively in a range of collaborative discussions with diverse partners, building on others' ideas and expressing their own clearly.	
CCSS.ELA-L.SL.4.4	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.	
CCSS.ELA-L.SL.4.6	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.	
CCSS.ELA-L.W.4.9	Draw evidence from literary or informational texts to support analysis, reflection, and research.	

professional development include tips for teacher "talk moves" [129], 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 [116]).

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 Figure 3.2).

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)

Figure 3.2: Computer Science Language Functions

3.5 Methods

This chapter builds the same research practice partnership described in chapter two of this dissertation between the University of California, Irvine and practitioners from the Orange County Department of Education and the Santa Ana Unified School District (SAUSD). As mentioned above, SAUSD has high percentages 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 research project, in this chapter 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 [91], paying specific attention to instructional practices that are effective for engaging multilingual students in STEM [27]. 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.

3.6 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. 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.

3.6.1 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 [169]. 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 [115]. 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.

3.6.2 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 [96] 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 3.2), 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 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 [77].

3.7 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 [20, 30], little research focuses on the instructional strategies that meet the specific needs of multilingual students [94].

Table 3.2: Audio Transcript of Jenny Teaching The Most Magnificent Thing

Speaker	Audio Transcript
Wendy:	(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)
Student 4:	A robot
Student 5:	A car...
Wendy:	(Wendy 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
Wendy:	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 8:	So she looked at all of her work that she thought was wrong.
Wendy:	And what did she notice about all of those things?
Student 3:	There were pieces that she liked.
Student 9:	There were the right pieces that she made.
Wendy:	So she had to do what? To her thinking? She had to do what to her thinking?
Student 10:	She had to look at her invention
Student 11:	Think more
Student 12:	Think about her problems so that she could fix them
Student 3:	Rethink her model
Wendy:	And what happened at the end? 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.
Wendy:	Think about that last page. Was it perfect?
Whole Class:	No!!
Wendy:	But did it do the job?
Whole Class:	Yes!!!

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 [27]. 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 [96]. 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 [96] 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.

3.8 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.

Chapter 4

Teaching computational thinking to multilingual students through inquiry based learning: A cross-case analysis

4.1 Chapter Synopsis

This chapter analyzes teacher enactment of inquiry-based learning during the implementation of an upper elementary computational thinking curriculum. I explore how teacher approaches to inquiry appear to support or constrain multilingual students' development of computational thinking skills and computer science identities. We adopt a cross-case mixed-methods design to collect data from five teachers and 149 students including detailed field notes, teacher interviews, computational artifacts, and student identity surveys. Through analyses of teacher moves, findings indicate that teachers adopt different approaches to inquiry that can be indexed along a continuum ranging from open to closed. Patterns in student data reveal that more structured inquiry-based approaches appeared to develop stu-

dents' computational thinking skills and computer science identities. This study helps to identify learning environments that increase participation for multilingual students across cognitive and affective dimensions, and has informed the next iteration of our curriculum.

4.2 Introduction

Considerable effort has been dedicated to integrating computer science (CS) into K-12 education for students who are traditionally underrepresented in STEM (e.g., women, students of color, students with disabilities). For example, the White House's 2016 Computer Science for All (CSforAll) initiative seeks to equip all K-12 students with the computing and computational thinking skills necessary to become creators, and not just consumers, of technology [185]. To help realize this goal, the National Science Foundation developed the CSforAll program which focuses on developing research-practice partnerships (RPPs) that foster the types of theory and practice needed to bring computer science and computational thinking to all students in K-12 schools. With a focus on CSforAll, educational policy makers and stakeholders have shifted their attention to developing CS pedagogy and materials that meet the needs of diverse learners.

While efforts to combat underrepresentation in computer science education have been numerous and laudable [79, 110], little attention has been paid to broadening participation for multilingual students ([95, 198], or those who speak a language other than English at home. This is especially important for the large and growing Latinx population–which grew from 9 million (6% of U.S. population) in 1970 to 59 million (18% of population) in 2017, and is projected to reach 132 million (30% of population) by 2050, but is seriously underrepresented in CS education and achievement. For example, in California, the site of this study, Latinx students constitute 54% of the K-12 population, but only 22% of advanced placement CS test takers [125]. A number of important obstacles hinder CS study for Latinx students, includ-

ing reduced access to home computers or family members who are knowledgeable about CS [168], lack of Latinx role models in CS whether through direct experience or through media representations [202], fewer course offerings in CS in Latinx-neighborhood schools [125], and decontextualized and individualized methods of CS instruction that are not a good match for the cultural values of Latinx students and families.

Inquiry-based learning has shown particular promise for engaging culturally and linguistically diverse students in STEM education [27]. Given the efficacy of inquiry-based learning for raising science achievement for multilingual students [68], inquiry-based approaches may also be effective for engaging these students in computer science education. Despite the promise of inquiry learning approaches, it remains unclear as to whether more structured or open inquiry approaches are more effective for engaging multilingual students in computer science. Proponents of open inquiry argue that the freedom to construct and conduct investigations develops students' higher order thinking skills, disciplinary knowledge, and inductive methods of inquiry [109]. Those who promote structured approaches claim that providing students systematic methods for conducting investigations supports the development of content knowledge, scientific skills, and a nuanced understanding of the discipline [159]. Structured approaches are further thought to prevent lost opportunities that arise from students getting stuck due to minimal guidance [194].

The purpose of this study is to analyze teacher enactment of inquiry-based learning during the implementation of an upper elementary, computational thinking curriculum, and to explore how teacher approaches to inquiry appear to support or constrain multilingual students' development of computational thinking. Whereas there is a plethora of research on inquiry learning in the domain of science [47, 207], few studies have analyzed how inquiry-based learning can be applied to computer science education. We adopt a cross-case mixed-methods design to collect data from five teachers and 149 students including detailed field notes, teacher interviews, computational artifacts, and student identity surveys. Through analyses

of teacher moves, we find that teachers adopt different approaches to inquiry that can be indexed along a continuum ranging from open to closed. Patterns in student data revealed that more structured inquiry approaches appeared to develop students' computational thinking skills and computer science identities. Findings from this study are being used to add more structured inquiry approaches to the next iteration of our curriculum including integrating USE/MODIFY/CREATE models into lessons and applying metacognitive strategies from reading research to students' programming activities.

Research Questions:

1. In what ways, if any, did teachers endeavor to teach computer science as inquiry to multilingual students in their classrooms?

2. How do differences in teachers' approaches to inquiry appear to support or constrain students' development of computational thinking skills and computer science identities?

4.3 Theoretical Framework

4.3.1 Inquiry-based Learning

Inquiry-based learning involves engaging students in authentic scientific practices and methods for the purpose of constructing knowledge [104]. Through student engagement in exploration, experimentation, and hands-on activities, inquiry-based learning provides a powerful mechanism for providing authentic contexts for language use [44]. During open inquiry-based learning, students develop questions and participate actively in open ended interrogations to discover and construct new knowledge [44]. During structured inquiry, teachers model methods and procedures for conducting investigations [207]. Inquiry-based learning emphasizes problem solving and students apply multiple problem solving approaches, practices,

and skills as they conduct their investigations [150]. Since the mid-1990s, teachers have been encouraged to better meet the needs of culturally and linguistically diverse learners by integrating inquiry-based approaches to make instruction more engaging, concrete, and meaningful [98, 45, 166].

Bybee [31] drew from constructivist approaches to learning to construct the "5E's" model, which includes five indicators of inquiry-based instruction: Engage, Explore, Explain, Elaborate, and Evaluate. Pedaste et al. [149] conducted a systematic literature review identifying key features of inquiry-based learning and synthesized a framework incorporating all elements of inquiry that persisted across models. They arrived at five stages: orientation, conceptualization, investigation, conclusion, and discussion [149].

4.3.2 Inquiry-based Learning and Computational Thinking

This general framing of the phases of inquiry relates inquiry to the scientific method, requiring a measure of reconsideration in its application to the field of computer science. Notably, scientific inquiry focuses on identifying and evaluating hypotheses to understand a set of principles governing the physical world. This objective contrasts with inquiry in computer science, which focuses on constructing and testing logical processes to address an abstract computational problem. Instruction in computer science then seeks to develop a set of skills, practices, and dispositions collectively referred to as computational thinking, representing an ability to formulate thoughts and questions for interpretation by a computer to achieve desired results (Wing, 2006).

An inquiry-based approach to computer science education must account for the myriad distinctions between the computational environment and the physical environment. We merge the five phases discovered by Pedaste et al. [149] with the 5 E's model [31] and use it as an analytic framework for understanding our field note data to better understand how

teachers enact the inquiry approach during the teaching of computer science lessons.

The first phases of both models, Orientation and Engagement, directly apply when beginning computer science lessons in the classroom. For example, "unplugged" activities that do not rely on using computers can introduce general concepts relating to computational thinking. These activities leverage students multimodal and everyday sense making abilities to stimulate interest in a topic or concept and prepare them to engage in the material, without directly incorporating computational technology.

The exercise of conceptualization in computer science, however, does not directly relate to developing research questions or hypotheses. Instead, computer science instruction typically involves illustrating abstract concepts such as variables, algorithms, and loops through discussion and reflection on their application to a specific problem. Through this exploration, students develop a frame for the problem at hand, in a manner similar to forming hypotheses guiding scientific study. However, this frame serves to characterize the solution to an abstract problem rather than the empirically observable properties of a physical phenomenon.

Scientific investigation then tests a student's hypotheses by contrasting observed evidence with their expectations. In computer science, investigating the viability of a proposed solution incorporates exploratory, explanatory, and elaborative steps in solving the problem. Through testing these solutions and debugging their implementation, students identify how to revise their conceived solution and adjust their programs accordingly. The iterative process of testing and debugging has some parallel to the cycle by which scientific hypotheses are revised. In so doing, investigative testing and debugging for computational solutions requires each of the last three E's. Students evaluate whether the program achieved its desired ends, explain why bugs prevented the computer from achieving these ends, and elaborate on the solution to resolve these bugs.

Conclusions from a computational thinking exercise evaluate the extent to which a proposed

algorithm or programmatic protocol accurately addresses the problem. This characterization mirrors the conclusions drawn from scientific exercises evaluating hypotheses' validity in light of empirical observation. During the conclusion phase, students typically apply their new knowledge of computing concepts, either through additional unplugged activities or through the exploration of computer programming environments. Conclusions for inquiry-based computational thinking, then arise from evaluating the efficacy of the program.

Finally, in accordance with both models, students engage in discussion throughout each cycle and evaluate their understanding through reflection-on-activity [174] and reflection-in-activity [58]. Discussion in inquiry-based computer science relays the approach used to solve the problems. We examine teacher observation data to investigate how learning takes place during each of the phases described above to characterize the types of inquiry that take place during computer science lessons.

4.3.3 Types of computer science inquiry

In its implementation, inquiry-based learning can be structured or unstructured based on teaching and learning goals of a lesson or unit of instruction [149]. Windschitl [207] conducted a multiple case study investigating how teachers perceived and enacted inquiry in their science classrooms and developed a continuum of inquiry demarcated by the degree of freedom students have in developing and conducting investigations (See Figure 1). At the structured end of the continuum lies confirmatory experiences, in which students are provided a systematic method for authenticating scientific principles. Next to confirmatory experiences lies structured inquiry, in which the teacher presents a scientific concept, question, or hypothesis and students are prescribed a procedure for exploring it. The next level is guided inquiry, where the teacher provides a problem to be solved but leaves the method of investigation up to the students. The most independent form of inquiry is open inquiry,

in which students identify their own concepts of questions and devise their own methods of investigation.

Figure 4.1: Inquiry-based Continuum Based on the Degrees of Freedom in Conducting Investigations

We use this continuum to characterize participants' approaches to teaching computer science lessons to multilingual students. We then draw from these findings to investigate whether patterns in student data can be plausibly explained by teachers' differing approaches to inquiry learning.

4.3.4 Inquiry-based Learning in STEM and Computer Science Education

Research on engaging students in Science, Technology, Engineering, and Mathematics (STEM) through inquiry based learning has been well established. A recent meta-analysis conducted by Estrella at al. [68] examined the effectiveness of inquiry instruction in increasing STEM achievement for elementary language learners. An analysis of 26 articles indicated that inquiry based instruction produced significantly greater results on measures of science achievement than traditional instruction. Furthermore, elementary students who participated in a blended program that integrated linguistic scaffolding with science inquiry-based unit plans showed statistically significant increases on California English Language Development Test (CELDT) and California Standards Test for English Language Arts scores compared to students in a traditional program [215].

A growing body of research on using inquiry to teach computational thinking demonstrates benefit for diverse learners [201, 17], however little research specifically focuses the types of inquiry approaches and levels of support that develop computational thinking for linguistically diverse students. Reiser [160] acknowledges the potential of inquiry to provide authentic learning contexts for students, but also articulates the challenges inherent to inquiry learning. For example, students need to acquire sufficient foundational knowledge to conduct investigations, and often focus on finding the right answer or solution to a problem as opposed to identifying the principles underlying answers. To ameliorate these issues, he proposes presenting more structured problem solving activities and problematizing subject matter to promote deeper understanding of content.

4.4 Overview of the Computational Thinking Curriculum

Researchers worked collaboratively with teachers to adapt an existing grade three through five curriculum created by Computer Science in San Francisco, a path-breaking initiative funded by salesforce that seeks to normalize computer science education in the San Francisco Unified School District from PreK-12. The curriculum was adapted to meet the needs of the participating district's culturally and linguistically diverse students. This was achieved by 1) integrating inquiry-based learning approaches, 2) aligning the curriculum with computer science and literacy standards, 3) developing linguistic scaffolds, and 4) providing culturally responsive pedagogy and materials.

First, researchers and teachers aligned materials with the Common Core State Standards for English Language Arts (ELA), and the California Department of Education English Language Development (ELD) Standards. Researchers and teachers then developed linguistic

frames to scaffold both the academic language related to computer science concepts as well as the functions of social interaction. To integrate inquiry based approaches, we utilized the "5 E" model of inquiry to guide unit development. Bybee [31] drew from constructivist approaches to learning to construct the "5E's" model, which includes five indicators of inquiry based instruction: Engage, Explore, Explain, Elaborate, and Evaluate. While we integrated the phases of inquiry into the curriculum, we encouraged teachers to use their own judgement when determining the level of structure necessary to meet students' needs. Finally, the Research-Practice Partnership paid special attention to integrating computational thinking and literacy development. Culturally responsive stories depicting diverse characters who pioneered the computer science and engineering fields were selected to make the content relatable to students.

4.5 Method

4.5.1 The Current Study

The context of this study involves the same research practice partnership described in chapters two and three of this dissertation between the University of California, Irvine, the Orange County Department of Education and the Santa Ana Unified School District (SAUSD). This network functioned through principles of Design-Based Implementation Research (DBIR), designing interventions to implement, study, and refine alongside the county and district. As mentioned above, SAUSD, among US school districts, has near the highest percentage of low-income students (91%), Latinx students (96%), and English learners (41%, higher in elementary grades, 63%).

We utilized a convergent, mixed-methods (Morse, 2003), cross-case study (Merriam, 2009) design to explore how teachers use inquiry-based approaches to implement a computational

thinking curriculum designed for linguistically diverse students. Cross-case designs are increasingly applying mixed methods approaches to study complex settings in which multiple researchers work to better understand phenomena [49, 48]. Utilizing a cross-case study design provides 1) the ability to cluster groups that share emerging patterns and order them along specified dimensions [133]; and 2) greater transferability of results as the number of cases increases [57].

4.5.2 Sampling Procedures

Context

This study took place in five upper elementary (grades 3-5) classrooms across the SAUSD. Student demographics at the classroom level broadly mirrored those at the district level.

Participants

A total of seven teachers and their classrooms were originally selected in the partnership program based on their prior experience and interest in teaching computer science to upper elementary students. Six of the seven continued the project, with one of the six deviating substantially from the original curriculum. The remaining five teachers and their classrooms constituted the sample for this study. All the students in their classes (total N=149 participated in the project and thus were part of the study.

4.5.3 Data Collection Methods

The participating teachers piloted the year long, five-unit computational thinking curriculum in their classrooms once a week for a lesson duration of fifty minutes.

Data Sources

Detailed field notes. For each field observation, two researchers took detailed field notes on teachers' instructional moves, students' interaction, and computing tasks and activities. The two researchers then met to compare and refine notes. Through multiple iterations, a framework was devised to answer our first research question. The framework addressed how the phases of inquiry were being enacted, focusing on the degree of freedom students were given during their investigations.

Teacher interviews on inquiry-based learning. The McGill Inquiry Teacher Short Interview (MITSI) protocol was used to interview the five participating teachers in their classrooms. Interviews were administered at the end of the school year and lasted approximately 20-25 minutes.

Student computational thinking outcomes. A rubric for scoring Scratch projects developed by SRI International was utilized to assess students' 1) overall proficiency in programming, 2) user experience, and 3) the use of coding and computational thinking constructs [16]. The overall proficiency category contains items that measures program correctness, code readability, and general impressions of novelty, engagement, and complexity. The user experience category contains items that measure user interactivity and the use of motion, media, and special effects. Finally, the coding and computational thinking constructs category measures the frequency and use of computational thinking concepts such as variables, loops, conditionals, operators, message passing, methods, complex data structures, program initialization, and program termination. Each item on the rubric is rated on a scale from zero to three as follows: 0 = Lacking basic proficiency, 1 = Emerging proficiency, 2 = Proficient, 3 = Exceeding grade-level proficiency. The researchers conducted interrater reliability checks on the rubric data. After each scorer completed ten projects, interrater-reliability ranged from 75% to 80%.

Student identification with computer science. This study drew from the "Is Science Me?" [9, 10] survey to measure the following: science and computer science activities, students' attitudes towards computer science, students' ability beliefs, perceptions of support for computer science interests from family and friends, and perceptions of academic support from family and friends. The constructs were grounded in research on the roles of family support [76], school experiences [144], and self-perceptions [206].

4.5.4 Results

Detailed field notes and teacher interviews. The generation of codes and categories in this study is situated within a procedural, deductive, frame of analysis [21, 112]. In this approach, top down coding was used to discover how teachers used inquiry-based learning principles to make the computational thinking curriculum accessible to multilingual students. This process consisted of reading the data multiple times to categorize inquiry learning phases and subcategories within the merged phases proposed in our theoretical framework. Codes and categories were then compared within and across each case to determine the types of inquiry (i.e., open, guided, structured, confirmatory being perceived and enacted in each class.

Computational thinking. A sum score for each category (e.g., overall proficiency, design mechanics, user experience) was calculated and z-score transformed. A one-way ANCOVA with post-hoc Tukey HSD test was conducted to examine whether there was a statistically significant difference among the teachers on the computational thinking criteria for the end-of-unit projects, controlling for student background information (i.e., computer access and parental education). Assumptions of normal distribution and homoscedasticity were met. When the ANOVA tests indicated that the scores across classes were significantly different, this study performed pairwise t-tests with Bonferroni and Holm adjustments to examine

which class was substantially different from the other classes.

Student identification with computer science. The survey mostly included three-point Likert scale items. This study calculated internal consistency using McDonald's omega instead of Cronbach's alpha, as Cronbach's alpha may be less accurate when data come from ordinal items with few response options [72]. The ω gives the variance proportion accounted for by a general factor (McDonald, 1981). The constructs improved their internal consistency from pre to posttest: attitudes towards CS (pre $\omega = .59$, post $\omega = .68$), perceptions of support from family and friends (pre $\omega = .67$, post $\omega = .74$), perceptions of support for CS interests from family (pre $\omega = .55$, post $\omega = .78$), perceptions of support for CS interests from friends (pre $\omega = .70$, post $\omega = .81$), science and CS practices (pre $\omega = .38$, post $\omega = .44$), and ability beliefs (pre $\omega = .44$, post $\omega = .67$).

A confirmatory factor analysis using polychoric correlations matrix was conducted based on the theorized constructs on the pretest and posttest datasets. Both datasets showed good model fit according to the guideline in Kline (2011): CFI and TLI higher than or equal to .90, RMSEA smaller than .05, SRMR smaller than .08. The fit indices for the pre-test were $2(85) = 162.68$, p = .07, CFI = .94, TLI = .92, RMSEA = .04 [.00, .05], SRMR = .07, and for the post-test were $2(85) = 169.22$, p = .03, CFI = .96, TLI = .94, RMSEA = .04 [.01, .06], SRMR = .06.

Because the survey items were ordinal and did not approximate a normal distribution, this study performed the Wilcoxon matched-pairs signed-rank test to measure the changes from pre to posttest for each survey item for each teacher.

4.6 Results

Based on classroom observations and detailed field notes, it became apparent that teachers enacted the curriculum differently across the five classrooms. The first four teachers used inquiry in different points along the continuum mentioned in the theoretical framework of this chapter, with Ellen using open inquiry, Juanita using guided inquiry, Jenny using a combination of guided and structured inquiry, and Helen using confirmatory experiences. The fifth teacher, Sue, did not use inquiry-based instruction and instead adopted a direct, explicit approach to teaching computer science content to her students. What follows are descriptions of teaching episodes for each classroom that are representative of how teachers conducted inquiry in their classrooms.

4.6.1 Open Approaches to Inquiry

Two of the teachers, Ellen and Juanita, exemplified open inquiry approaches in mentoring their students through various aspects of computer science research. On a typical day, both teachers would orient students to computational thinking concepts through activities structured around focal phenomena, and facilitate collaborative sense-making of key computational thinking concepts and practices. During investigation, Ellen often promoted independent learning in her classroom, acting as a facilitator of the research process by equipping students with the resources and strategies necessary for conducting open-ended investigations.

Ellen openly expresses her views of computer science learning as a research process in which students seek out the resources necessary to solve complex problems.

> Ellen: When you get stuck, we have resources. I am not the greatest resource because I am learning with you too. I can guide you to resources. Your peers are

a resource... When you get stuck, we have resources...Am I your only resource?

Students: No

Ellen: You know, many of you have learned that I am not the greatest resource. I'm not wanting to be. Why, because I'm learning this with you too. Okay. So I can kind of guide you in how to be resourceful. I'm finding more and more places that I can get help when I need it and that's what you need to do as scholars.

Ellen describes herself as a being a guide for her students, using herself as a model to illustrate the types of habits (i.e., being resourceful, help-seeking) her students can engage in as scholars. To this end, she disrupts her traditional role as teacher to create a more horizontal, symmetrical space in which students and the teacher co-navigate computer science research. She provides the learning environment and resources necessary for students to ask and answer complex problems, and steps aside to facilitate scholarly activity.

Juanita similarly promotes independence during the investigation phases of inquiry, but unlike Ellen, models methods of problem formulation. She also disrupts the direct instruction model by encouraging students to negotiate their own learning among peers before coming to her for answers. For example, in the excerpt below Juanita modeled the first steps in a shape drawing activity designed to teach algorithms. Students were presented with written steps for drawing shapes and a picture with the desired visual outcome (i.e. a picture of a house). However, the steps did not match the shape, that is, there were errors embedded in the steps and students were tasked with debugging the algorithm so that the steps matched the desired outcome.

Juanita: Okay, again let's look at this, who could tell me where my equilateral triangle is

Roxanne: On top of the blue square

Juanita: Remember, equilateral means it's what on all sides.

Class: Equal.

Juanita: Equal. Very cool. Okay, so check that it's lined up with the top of the blue square. Ok, so do you guys see it now?

Class: Yes

Juanita: Now talk to your table, your elbow partner. And I want you to go through...I want you to figure out with your partner the rest of the steps and let me know what you think the bug is. Once you've figured out what the bug is, I don't want to hear any ah ah ah's or ooh ooh ooh's. Figure out what it is before you raise your hand.

In this teaching episode, Juanita defined the scope of the problem for students and modeled the first few steps for solving it, then encouraged students to rely on their peer networks to finish debugging the algorithm. To this end, she established a peer learning community in which students discuss debugging techniques with their classmates before they ask the teacher. As students were provided with a problem, but not explicitly given a procedure for solving the problem, the above example supports the guided inquiry model. After students worked to debug the sample algorithm, students were assigned to individually create their own algorithm and drawing using only their peer networks for support, further supporting the guided inquiry hypothesis.

4.6.2 Structured Approaches to Inquiry

Two of the teachers, Jenny and Helen, tended to teach inquiry learning in a structured manner by 1) providing methods for formulating and solving problems, or 2) demonstrating and confirming key computational principles. Jenny utilized a combination of guided and

structured inquiry to teach computer science to her students. She typically opened with a guided inquiry lesson and then moved to more structured approaches when students had difficulty accessing abstract concepts. In the example below, Jenny initially used the guided inquiry approach to develop students' understanding of sequence and repetition. Students were provided with multiple steps of a dance and tasked with working in teams to conceptually map the dance, using loops so as to not articulate each single step. While Jenny identified the scope of the problem for students, she did not provide with them methods for solving the problem.

> Jenny: In your team, I want you to identify the actions in this dance... write a computer code, if you were to tell someone how to do this dance, what would you do, are there repeated actions, how many times do they repeat. What would an algorithm, a code for this, look like?

In this example, students were given more freedom to develop and conduct their own investigations and the teacher focused primarily on facilitating the learning process and detecting student issues as they arose. As Jenny focused in on students' needs, she moved from guided to structured inquiry to create more efficient learning environments. For example, students had difficulty characterizing the dance on their first attempt, likely because they had not committed the sequence of moves to memory. In response, Jenny used a variety of techniques to respond to this need including replaying the dance, refocusing students' attention to counting and naming the moves, using dialogic questioning to facilitate students' recall of the dance, and physically enacting the dance before the class. Each of these instructional moves points toward a more structured inquiry approach, that is, Jenny modeled alternative methods for students to investigate the key computational concepts. Helen provided structure for her students through confirmation experiences designed to teach computing concepts. Helen's typical investigation was highly structured and relied primarily on teacher modeling and scaffolding techniques to facilitate knowledge acquisition. To illustrate a simple

example, Helen drew two sprites, or characters in the Scratch interface, on the whiteboard, prepping the class to discuss what parallelism means.

Helen: What do I do if I want the sprites to do the same thing at the same time?

Next, Helen drew two columns, one for sprite 1 and one for sprite 2 and added blocks to each column. Then, she manipulated blocks in a variety of ways and asks students to make predictions.

Helen: What's going to happen?

To test the concept, Helen hit a mock flag button to start a "test run" of parallel commands. When sprites 1 and 2 correctly executed their commands, students verified their understanding of how to use commands in Scratch to make two sprites take actions simultaneously.

Helen: It worked!

Upon checking student understanding, Helen added additional layers of complexity to teach more advanced computational concepts. For example, she modeled more complicated multi-step commands for each sprite to execute simultaneously, except in this case the sprites are doing two different things at the same time (i.e., sprite 1 walks forward, sprite 2 jumps up and down). Furthermore, she purposefully included errors in her mock code to engage students in debugging during concept development.

In the above scenario, students were presented with a variety of scenarios in which parallelism could occur, and then these scenarios were verified by 'test runs.' Although she increased the complexity of her examples, students were not provided the opportunity to generate or conduct investigations on their own. Instead, they engaged in confirmatory experiences of key computational principles.

Direct Approaches to Teaching

Sue did not take an inquiry-based approach to teaching computer science to her students. Instead, she focused on rhythm and periodicity within her classroom, ensuring that students had access to stable routines and durable infrastructure to support knowledge acquisition. This allowed her to manage interaction and bring about predictability of sequential classroom activities and students' behaviors.

Sue predominantly used the mirroring technique to teach computer science concepts to her students. In the example below, she taught the concepts of sequence and order, initiating the mirroring activity, in which she would say "mirrors on", and the students would imitate her words and actions:

> Sue: Okay. I liked what you said about the events, but when we're talking about sequence of events, what does that mean? Is it, are we talking about groups or am I talking about order?
>
> Class: Order
>
> Sue: Right? Mirrors on! Sequence (Sue waives hand motions, class repeats) is order (Sue waives hand motions, class repeats) or sequence is the order of events.

In this excerpt, Sue took a direct, explicit approach to teaching, enforcing targeted stimuli (i.e., teacher models specific motions) and response (students mimic teacher's motions) behaviors, coupled with repetition to teach key concepts. She also reinforced correct verbal and motor associations with a clip up classroom management technique, using positive reinforcement as an example for the class of what types of behaviors are preferred.

4.7 Student Outcomes

To better understand students' previous experience with computer science, we provide a breakdown of student demographics, disaggregated by classroom, based on students' access to computers, parental education, and parents' exposure computer science careers.

4.7.1 Student development of computational thinking through programming in Scratch

First, we combined students' scores on items using the selected SRI rubric scale. When examining teachers' overall combined average rubric scores for complexity, design mechanics, user experience, and use of computer science constructs, we see that Helen and Jenny's students performed better overall (See Figure 4.2) and that all students performed close to Emerging proficiency.

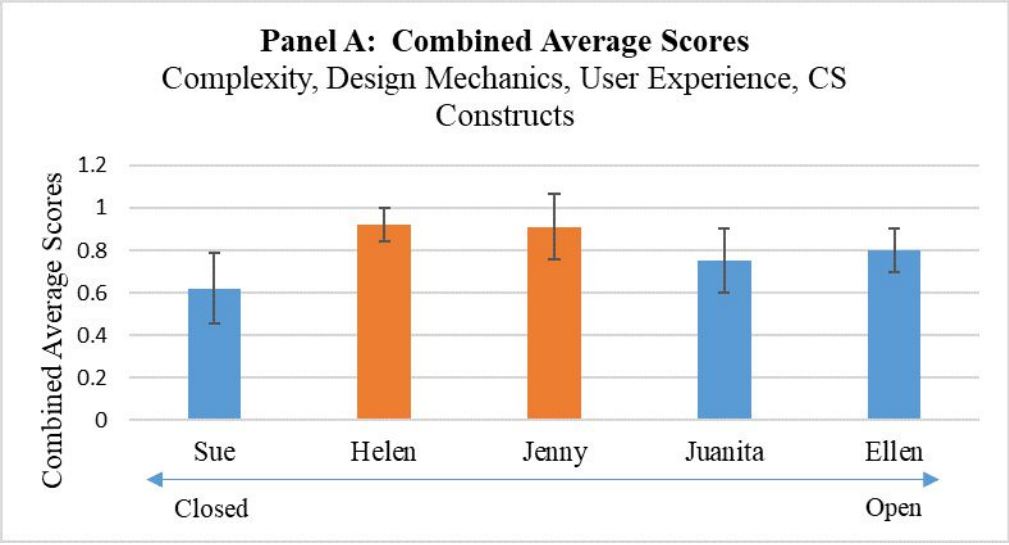

Figure 4.2: Students Combined Scores for Complexity, Mechanics, User Experience, and Computer Science Constructs, by Class

Furthermore, we counted the types and frequency of blocks used according to number of sprites, conditionals, variables, loops, arithmetic operators, Boolean operators, and methods

present in the project. Averaging overall scores for each teacher in this category, we see that Helen and Jenny's students on average used a higher frequency of the aforementioned blocks than other students (See Figure 4.3).

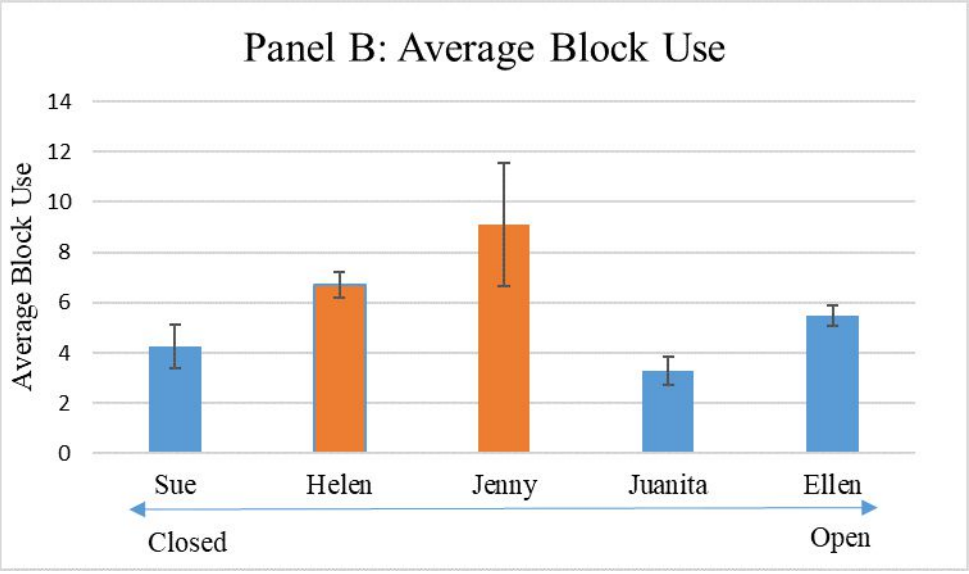

Figure 4.3: Frequency of Block Usage According to Computational Thinking Constructs, by Class

Results from the one-way ANCOVA showed that Helen's and Jenny's class performed substantially better in several criteria (See Figure 5.1). There were significant differences among the classes in overall complexity, $F(4, 107) = 17.33$, $p < .01$; user experience, $F(4, 107) = 5.27$, $p < .001$; CS constructs, $F(4, 107) = 9.11$, $p < .001$; and counts of different types of Scratch blocks, $F(4, 107) = 4.58$, $p < .01$, after accounting for home computer access, mother's education, and father's education. Tukey HSD test revealed that there was a significant difference in the overall scores for Ellen and Sue ($p < .001$), and Jenny and Sue ($p < .01$). For user experience, there was a significant difference between Helen and Ellen ($p = .03$), Juanita and Sue ($p < .01$) and Helen and Sue ($p < .001$). For CS constructs, there was a significant difference between Helen and Ellen ($p < .001$), Helen and Juanita ($p < .001$), Ellen and Sue ($p < .001$), Sue and Juanita ($p = .01$), and Helen and Jenny ($p = .02$). For block use, there was a significant difference for Juanita and Helen ($p = .01$), and Juanita and Jenny ($p < .01$).

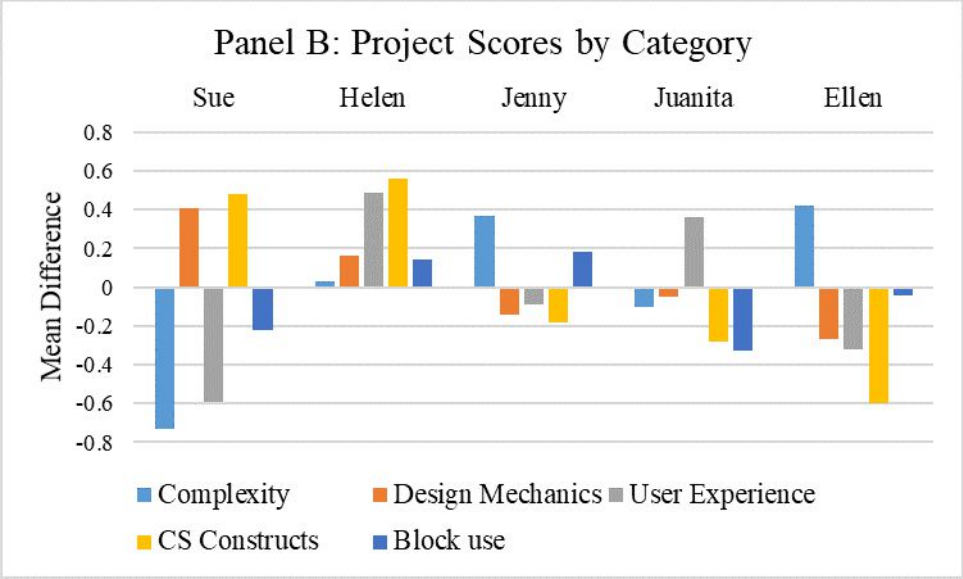

Figure 4.4: Differences in Project Scores by Category, by Class

4.7.2 Student identification with computer science

There appeared to be positive changes from pre to posttest in three classes (Helen, Jenny, and Sue) in terms of students' ability beliefs, perceptions of support for computer science interests from family and friends, and perceptions of the usefulness and importance of computer science (See Figure 4.5). The effect size ranged from medium to large, Cohen's d = (.46, 1.07).

4.8 Discussion

The idea of teaching science as inquiry has been well supported by research [55, 44] and has been found particularly effective for engaging English learners in STEM [68]. Recently, researchers have called inquiry learning into question [32**?** , 189], finding that explicit instructional approaches better support learning and transfer [106, 108, 213]. This is one of the first studies to investigate how different approaches to inquiry support or constrain multilingual students' development of computational thinking skills. Preliminary findings indicated that

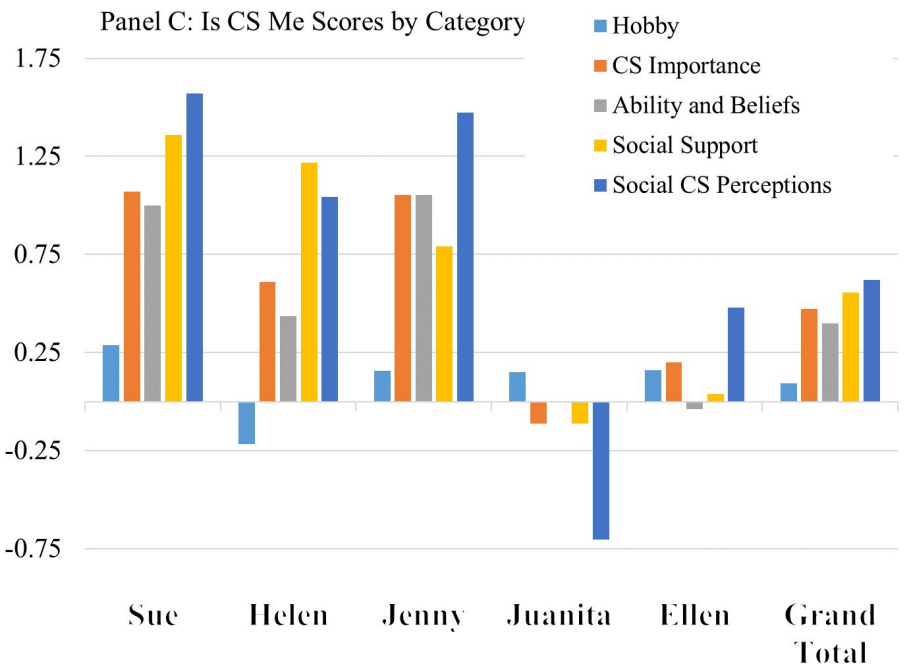

Figure 4.5: Differences in Identity Survey Scores by Category, by Class

more structured forms of inquiry appear to better support multilingual students in engaging in and identifying with computer science.

4.8.1 Overview of Findings

A primary aim of this study was to explore the question: "In what ways, if any, did teachers endeavor to teach computer science as inquiry in their classrooms?" One of the ways to gain insight into how teachers used inquiry approaches was to analyze how teachers enacted computer science lessons in their classrooms. By exploring the phases of inquiry that emerged from the lessons, we were able to piece together the ways in which teachers approached inquiry and index these approaches as being more open or closed along an established continuum. While the phases [31, 149] and types [207] of inquiry in science education have been well established, our study provides considerable insight into how the types of inquiry pertain to the discipline of computer science. These results extend our knowledge of the ways

in which computer science can be taught as inquiry. Furthermore, the theoretical framework presented in this chapter, relating the phases of inquiry to computer science education, can be extended, refined, and used as a conceptual framework for analyzing classroom interaction.

An additional aim of this study was to explore how teachers' differing approaches to inquiry appeared to support or constrain multilingual students' development of computational thinking skills. Patterns in the data revealed three clusters of teachers, those whose teaching sequences revealed more open approaches, those whose sequences revealed more structured approaches, and those whose patterns displayed direct instructional approaches. For teachers who were indexed as being more structured along the inquiry-based learning continuum (Helen and Jenny), students tended to develop more sophisticated Scratch projects. We present several conjectures for this relationship. In these classrooms, teachers used modeling techniques to illustrate methodologies for solving problems, such as simulating algorithmic processes, physically enacting computational concepts, and incrementally increasing levels of complexity. Furthermore, these teachers provided worked examples to characterize conceptual and investigative principles, using strategies such as repetition, refocusing students' attention, and open ended questioning techniques to address student confusion. Studies have indicated that providing worked examples reduces the tax on working memory and opens up the resources necessary for learning [105]. Through targeted schema building, students draw from a conceptual foundation when addressing increasingly abstract problems, thereby building knowledge from prior understandings [186].

In this study, we also see that more structured and direct approaches produced better outcomes for diverse learners' identification with the field of CS. Although inquiry-based instruction engages students in authentic scientific practices, delivering unstructured inquiry without sufficient schema building and preparation can lead to disappointment and lost opportunities [22]. All four teachers who implemented inquiry-based approaches reported students getting lost, getting stuck, and jumping in without seeing the big picture. If we want

to broaden participation in computing, it is important to not only give students experiences, but to give them successful experiences. We were especially impressed the work from Estelle, herself a Latina who had substantial experience with the community she served and taught large number of multilingual students and students with disabilities. Further exploration into the methodological and incremental approach she used could uncover valuable strategies to broaden participation for linguistically diverse learners by meeting their cognitive and affective needs.

4.8.2 Limitations and Future Research

There are several limitations to this study. First, as this was an exploratory study that is part of a larger project aimed at revising, testing, and scaling instructional materials, we do not make causal claims about our findings. Each of the classes we observed had different compositions and grade levels and these factors could provide confounds to our claims. Furthermore, the data instruments we used to measure students' computational thinking skills may not be sensitive to the types of learning that took place in the open inquiry classes. In open inquiry classrooms, different types of learning and growth may take place, such as problem formulation, goal setting, planning strategies, and persistence. Future studies should design measures for capturing the types of learning and growth taking place in these classrooms.

Despite these limitations, this chapter poses several questions to the understudied area of teaching computational thinking to multilinguals students through inquiry-based learning. As this project represents the exploratory phase of a larger project, we are currently using these findings to integrate more structured approaches in the next iteration of our curriculum. This includes integrating the USE/MODIFY/CREATE model into our lessons and applying metacognitive strategies from reading research to students' development of compu-

tational thinking. To integrate more structure, the next phase of this project will modify the curriculum to integrate a CS instructional approach known as Use-Modify-Create [113, 205] in which students will first use existing programs, then work together to modify them, and finally create their own. Furthermore, during the use stage, we will incorporate an additional layer of scaffolding with a learning strategy borrowed from reading research known as TIPP&SEE [171], developed by the Computing for ANyONe (CANON) lab at the University of Chicago and faculty at Texas State University. TIPP&SEE is derived from the reading strategy THIEVES [123], and focuses students on using context clues to better grasp intended material. Students first read the title of the program and make predictions based on the title. Then they analyze the instructions to better understand the tasks they are asked to engage in. Next, students think about the purpose of the program to consider the learning goals of the activity. Finally, they play with the program to examine its characteristics and practice documenting their observations. Students are then tasked with looking inside the program to examine the sprites and the events controlling the sprites, and then they explore the code. During the explore phase, students are instructed to change features of the code, test the changes, and document the results, preparing them for the MODIFY stage of the USE/MODIFY/CREATE model. This new curriculum will be scaled to three school districts and tested using randomized control trial to formally examine the impact of structured approaches to inquiry on multilingual students' development of computational thinking skills.

Chapter 5

Discussion

Whereas California represents the nation's tech capital, diverse students in the greater Los Angeles area are often left in the margins of computer science (CS) education. My research project represents one of the first initiatives in the nation to develop and test a computer science curriculum that helps multilingual students succeed in computing. The curriculum employs what we know about linguistic scaffolding, culturally relevant pedagogy, and instructional design to help multilingual students and other diverse learners leverage their own and their community resources for greater success and entry into the field of computing.

Computational thinking skills have become integral to full participation in today's society [208]. Yet, very little is known about student engagement with computational thinking in formal learning environments, especially among students who speak a language other than English at home. Given that early exposure is foundational to developing interest in later CS careers, educators must understand how children engage with computational problem solving and the factors that drive student engagement. Furthermore, by examining the knowledge, skills, and attitudes towards computational thinking among culturally and linguistically diverse students, we will understand the various characteristics that play a role in the way

children engage with computing technologies.

5.0.1 Significance of the Research

This dissertation is one of the first major studies to systematically investigate and identify promising practices for teaching multilingual students computer science. It will contribute to our knowledge about culturally responsive CS instruction, the intersection between CS education and language and literacy development, and effective approaches for teaching CS in diverse elementary schools. In addition, my research project will help initiate a new line of research in the US on quality CS instruction for multilingual students. Based on prior research on Universal Design for Learning, it is expected that these practices will also benefit many other groups of learners, including students with limited literacy, those with reading disabilities, and speakers of non-standard English dialects [165].

5.0.2 Practical Implications

This research has impacted children from diverse backgrounds and spearheaded actionable changes in the field of computer science education. While there is no time dedicated to CS in elementary in the US, by the end of this curricular roll-out, all fourth-grade students in our participating district will receive a protected 50 minutes of weekly CS instruction that opens doors to future study and careers. In addition, this research has the potential to impact many other students across the country as other districts begin adopting our curriculum. Furthermore, each of the three studies in this dissertation provides recommendations for teachers, administrators, and educational policymakers on developing multilingual students' CS identities and literacy skills while providing instruction that develops their disciplinary identification.

5.0.3 Methodological Implications

To date, the limited amount of studies examining multilingual students' use, beliefs, and attitudes about computer science have either been theoretical or empirical, using only anecdotal examples to illustrate findings. The findings that the mixed methods data from this dissertation give us insight into multilingual students' learning of and identification with the field of computer science and the types of instructional practices that benefit diverse learners. This is important given that they are one of the fastest growing populations in the U.S.

It is important to note that the three studies in this dissertation are exploratory, and data are collected from pilot-level implementation. The pilot's goal was to develop research questions that could serve as the basis for testing hypotheses at the study's evaluation stage. To this end, researchers can use findings from this study to help identify theories related to developing multilingual students' CS identities. Researchers can further evaluate these theories in the form of hypotheses in larger-scale implementation, using experimental or quasi-experimental designs. In addition, investigators could gather demographic data on participating students in future studies. This data could shed light on how students from specific and intersecting marginalized groups (i.e., women, students designated as English learners, students with disabilities) develop CS identities compared to their traditional counterparts.

A major limitation of the first study in this dissertation is that there was no control group; future work using randomized control trials will lead to a better understanding of the effectiveness of the curricular implementation in developing multilingual students' CS identities.

The second study in this dissertation uses qualitative data to better understand how CT-ELA integrated curriculum simultaneously develops students' CT and literacy skills. Furthermore, the study's theoretical framework provides an analytic approach to understanding how teach-

ers can leverage students' existing literacy skills to develop their computational thinking skills and vice versa. Most studies that seek to understand the relationship between computational thinking and literacy frame computational thinking as a form of literacy. Only a few studies provide a framework for understanding CT and literacy integration that teachers can use to improve student outcomes. The qualitative data in this study offers moment-level explanations of how teachers use literacy to teach key CT concepts. Future research can look at students' CT, language and reading assessments to better understand how the CT-ELA integrated curriculum improves literacy outcomes for multilingual students.

The third and final study in this dissertation identifies the types of inquiry-based instruction that maximize CS learning for multilingual students. Findings indicate that structured approaches to inquiry-based instruction support multilingual students' development of computational thinking skills and CS identities. While quantitative findings from the cross-case analysis are correlational, they shed much needed light on the types of instructional approaches that benefit culturally and linguistically diverse learners during CS instruction. As researchers and practitioners developed the curriculum using Design-Based Implementation Research, findings from this study provided a roadmap for iteratively refining the curriculum to offer more structured approaches to CS learning. These include integrating USE/MODIFY/CREATE [113, 205] models that allow students to use and modify existing models before creating their own. At the USE stage, an additional layer of scaffolding is used called TIPP&SEE [171], which borrows from research on reading education [123]. TIPP&SEE has been incorporated into the materials to provide students opportunities to contextualize instruction while engaging more closely with lesson concepts.

5.0.4 Integrating Multilingual Students' Identities, Literacies, and Pedagogies

Though this study looks separately at issues of identity, literacy, and pedagogy for analytical purposes, these three topics are highly interrelated. Collectively, these dissertation studies contribute to an asset-based approach for engaging multilingual students in CS education. Multilingual students come to the classroom with rich and varied resources that can be leveraged through culturally and linguistically responsive pedagogical approaches. The figure below presents a theoretical framework for characterizing this relationship.

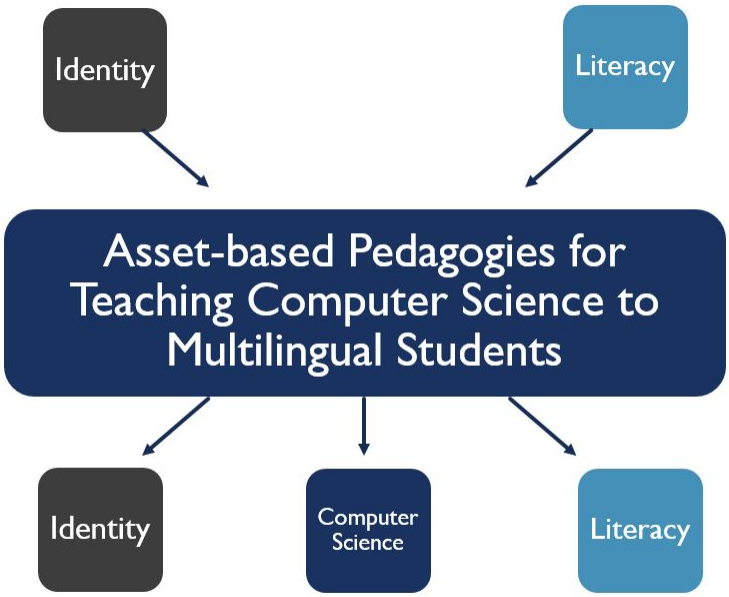

Figure 5.1: Theoretical framework for understanding asset-based CS pedagogy for multilingual students

Asset-based CS pedagogies leverage multilingual students' identities, languages, and literacies through culturally responsive curricula that provide structured learning opportunities for young learners. Multilingual students' emergent literacy skills and disciplinary identities represent inputs in the framework above. Asset-based approaches to CS pedagogy build on these inputs by making connections to multilingual students' rich resources while providing

the scaffolding necessary to avoid frustration and wasted opportunities that arise from poorly structured learning environments. In turn, this strengthens multilingual students' literacy skills, CS identities, and computational thinking skills, representing the model's outputs above.

Asset-based pedagogical approaches are critical to providing culturally responsive opportunities for multilingual students to succeed in CS while minimizing experiences of getting lost due to minimal guidance. The first study of this dissertation connects classroom learning to informal learning environments, which strengthens multilingual students" CS identities and contributes to their persistence in the face of challenges. Moreover, these informal learning environments afford a wealth of knowledge present in students' families, peers, and communities that can be leveraged to enhance classroom learning and disciplinary identification. To this end, drawing upon students' informal identities and emergent literacy skills through scaffolding strategies, such as teacher modeling and worked examples, has the potential to transform our understanding of how students develop disciplinary identities and literacy skills. Typically, disciplinary identities have been understood as the knowledge, skills, and attitudes necessary to become a competent actor in a given community of practice [26, 35]. However, there has been a recent shift toward understanding the linguistic and sociocultural factors that shape multilingual students' identity formation and contribute to language and content learning [34].

To this end, this dissertation contributes to the discussion on asset-based approaches for engaging multilingual students by viewing their budding identities and emergent literacies as levers for increasing their disciplinary identification and literacy development. However, these levers can only be utilized when instruction provides support and responsiveness that enables multilingual students to build on their existing resources to strengthen their CS identities and literacy skills. This approach is exemplified in the theoretical model in study two of this dissertation which explores the relationship between computational thinking and

literacy. This model adopts an asset-based approach to advance our understanding of how multilingual students' existing literacy skills can be leveraged to develop their computational thinking skills and vice versa [96]. Furthermore, many strategies used to develop multilingual students' computational thinking and literacy skills mirror the strategies used in the third study to provide structured inquiry to engage multilingual students in computer science education. The third study of this dissertation finds that structured approaches to inquiry-based learning support the development of computational thinking skills and CS identities in multilingual students. Taken together, these findings point to a model of instruction that provides the support necessary for multilingual students to build on their existing knowledge base to contribute to greater identity development and new forms of literacy. This model can be used as a framework for providing asset-based instruction in other subjects as well, such as math, science, and social studies.

In conclusion, this framing borrows heavily from Warschauer [203] and provides a model for understanding the iterative relationship between identity, literacy, and pedagogy. To explain, students' existing identities and literacy skills can be leveraged by teachers as resources during instruction. In this way, students' CS identities and literacy skills are also strengthened as a result of applying the asset-based CS pedagogies described in this dissertation. To this end, asset-based CS pedagogies serve to develop and enhance these resources for multilingual students. When applied with care, asset-based CS pedagogies can facilitate identity and literacy development for multilingual students. However, poorly delivered CS instruction risks reproducing inequities by stifling identity and literacy development, thereby perpetuating the marginalization of culturally and linguistically diverse learners.

5.0.5 Implications for Educational Leaders, Policymakers, Teachers, and Researchers

Implications for Educational Leaders and Policymakers

CS instruction should be introduced at a young age so that students can develop positive ability beliefs and identification with the field. These dispositions will increase the likelihood of student enrollment in future courses and persistence in the face of challenges. In addition, there are distinct synergies between CS and language and literacy development that contributes to multilingual student learning rather than distracting from it. Therefore, educational leaders and policymakers should seek to integrate computer science content into English Language Arts courses to improve both computer science learning and literacy development for multilingual students. This integration is especially critical in elementary grades, as instructional minutes allotted to STEM are extremely limited. Integrating CS into existing content has multiple affordances, including but not limited to providing systematic opportunities for CS learning, enabling students to make deeper connections between cross-curricular content, and increasing student motivation. Providing these opportunities prepares young learners to participate in a knowledge economy and equips them with the tools to become active participants in shaping technological advances.

Implication for Teachers

Findings from this dissertation highlight the pedagogical value of teachers making broader connections to students' existing resources. We provide several recommendations for bridging the gap between classroom learning and children's out of school experiences, including 1) leveraging students' personal and family backgrounds, 2) garnering parental support, 3) making connections between the CS profession and broader social issues, 4) positioning

students as experts, 5) engaging peer support networks, and 6) providing multiple points of connection between formal and informal learning environments. These recommendations are easy to implement and foster multilingual student CS identity development and persistence in the face of challenges.

We also list several strategies that teachers can use to mobilize students' existing literacy skills to foster computational thinking, including 1) activating prior knowledge, 2) using multiple questioning techniques, 3) providing direct instruction, and 4) providing language support. These strategies represent good teaching, not only for multilingual students, but for all students including students with disabilities, students of color, and gifted and talented students. We dig deeper by providing vignettes that illustrate how teachers leverage students' existing literacy skills to develop their computational thinking skills. We hope that these strategies and vignettes will provide practical examples of asset-based approaches to teaching integrated computational thinking and literacy lessons to multilingual students.

Finally, this dissertation finds that structured approaches to inquiry-based learning support the development of computational thinking skills and computer science identities for multilingual students. Therefore, integrating scaffolding strategies such as teacher modeling and worked examples into inquiry-based approaches to learning provides greater access to CS content compared to unstructured learning approaches. Taken together, these practical recommendations provide equitable learning opportunities for culturally and linguistically diverse learners.

Incorporating computational thinking into literacy instruction requires only a modest investment of time and effort by teachers. Several existing curricula integrate CS content with language and literacy, including the Elementary Computing for ALL [2] project developed by the University of California, Irvine and the University of Chicago. This curriculum offers culturally and linguistically responsive, structured CS and literacy lessons for students in grades 3-5. In addition, the Coding as Another Language [1] provides similarly integrated

coding and literacy lessons for students in grades 2-5. Teachers with little to no prior coding experience can deliver these age-appropriate, year-long curricula, which are highly accessible to culturally and linguistically diverse learners. Furthermore, they have also been shown to simultaneously increase students' computer science and literacy development.

Implications for Researchers

Although computer science has been typically considered a STEM subject, this dissertation points to the cultural, social, and linguistic practices involved in computing. To illustrate, STEM identity development has been typically construed as acquiring the knowledge, skills, and attitudes necessary to become competent in a technical subject. However, there has been a recent shift toward understanding the sociocultural and linguistic factors contributing to identity and literacy development. In this dissertation, identity formation is framed as sociocultural and linguistic participation in CS communities of practice [111]. As such, students who develop positive disciplinary identities develop competence and begin to assume the agency necessary to shape CS practices. Similarly, definitions of literacy have shifted from the notion of skills development (i.e. reading, writing, listening, speaking) toward the widespread dissemination of social practices [59]. As students actively engage in literacy practices, they acquire the agency necessary to contribute to new forms of literacy development. Understanding the sociocultural and linguistic factors that shape multilingual student engagement in computer science education helps researchers frame student participation in culturally responsive ways, increasing equity and access for marginalized learners.

5.0.6 Conclusion

Computing technologies have the potential to transform the modern educational landscape, providing students opportunities to become developers, not just consumers of technology

[185]. However, we must first understand how different students learn and identify with computational thinking and how designing and tailoring effective instructional materials help multilingual students succeed in mastering computational thinking. Given the dearth of data on multilingual student participation in CS, there has never been a more critical time to examine the factors that contribute to their success and entry into the field.

This dissertation highlights the challenges and opportunities associated with engaging multilingual students in computer science education. While further research is necessary to better understand this issue, these exploratory findings provide a foundation for the knowledge base establishing best practices to provide equitable opportunities for culturally and linguistically diverse students. By building on what is known, we can offer direction for educational leaders, policymakers, teachers, and researchers to better serve multilingual students in computer science education. While there is much work left to do, initiating this discussion propels us towards providing culturally and linguistically responsive computer science for all.

Bibliography

[1] Coding as Another Language, https://sites.tufts.edu/codingasanotherlanguage/. Accessed: 2022-04-29.

[2] Elementary Computing for All, https://www.elementarycomputingforall.org/team.html. Accessed: 2022-04-29.

[3] J. Adams. The fourth age of research. *Nature*, 497(7451):557–560, 2013.

[4] V. A. Aguayo. Life after the el label: Conversations about identity, language, and race. 2020.

[5] J. Ahn. *Computational Thinking in Children: The Impact of Embodiment on Debugging Practices in Programming*. PhD thesis, Teachers College, Columbia University, 2020.

[6] J.-H. Ahn, Y. Mao, W. Sung, and J. B. Black. Supporting debugging skills: Using embodied instructions in children's programming education. In *Society for Information Technology & Teacher Education International Conference*, pages 19–26. Association for the Advancement of Computing in Education (AACE), 2017.

[7] A. Allen-Handy, V. Ifill, R. Y. Schaar, M. Rogers, and M. Woodard. Black girls steaming through dance: Inspiring steam literacies, steam identities, and positive self-concept. In *Challenges and Opportunities for Transforming From STEM to STEAM Education*, pages 198–219. IGI Global, 2020.

[8] L. Archer, J. DeWitt, J. Osborne, J. Dillon, B. Willis, and B. Wong. Science aspirations, capital, and family habitus: How families shape children's engagement and identification with science. *American Educational Research Journal*, 49(5):881–908, 2012.

[9] P. R. Aschbacher, E. Li, and E. J. Roth. Is science me? high school students' identities, participation and aspirations in science, engineering, and medicine. *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching*, 47(5):564–582, 2010.

[10] P. R. Aschbacher, S. M. Tsai, et al. Is science me? exploring middle school students' ste-m career aspirations. *Journal of Science Education and Technology*, 23(6):735–743, 2014.

[11] J. Banks, K. Au, A. F. Ball, P. Bell, E. Gordon, K. Gutiérrez, S. Brice-Heath, C. D. Lee, J. Mahiri, N. Nasir, et al. Learning in and out of school in diverse environments: Life-long, life-wide, life-deep. 2007.

[12] P. Barmby, P. M. Kind, and K. Jones. Examining changing attitudes in secondary school science. *International journal of science education*, 30(8):1075–1093, 2008.

[13] B. Barron, C. K. Martin, L. Takeuchi, and R. Fithian. Parents as learning partners in the development of technological fluency. *International Journal of Learning and Media*, 1(2):55–77, 2009.

[14] A. C. Barton and K. Yang. The culture of power and science education: Learning from miguel. *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching*, 37(8):871–889, 2000.

[15] S. Basu, G. Biswas, P. Sengupta, A. Dickes, J. S. Kinnebrew, and D. Clark. Identifying middle school students' challenges in computational thinking-based science learning. *Research and practice in technology enhanced learning*, 11(1):1–35, 2016.

[16] S. Basu, R. Clopton, and C. Tate. Tfinal report: Year 5 sfusd stem learning initiative evaluation: Computer science artifact analysis.

[17] S. J. Basu and A. C. Barton. Developing a sustained interest in science among urban minority youth. *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching*, 44(3):466–489, 2007.

[18] P. Bell, C. Tzou, L. Bricker, and A. D. Baines. Learning in diversities of structures of social practice: Accounting for how, why and where people learn science. *Human Development*, 55(5-6):269–284, 2012.

[19] K. Bell-Watkins, T. Barnes, and N. Thomas. Developing computing identity as a model for prioritizing dynamic k-12 computing curricular standards. *Journal of Computing Sciences in Colleges*, 24(3):125–131, 2009.

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

[21] R. E. Boyatzis. *Transforming qualitative information: Thematic analysis and code development*. sage, 1998.

[22] J. D. Bransford, A. L. Brown, R. R. Cocking, et al. *How people learn*, volume 11. Washington, DC: National academy press, 2000.

[23] M. A. Bravo and G. N. Cervetti. Attending to the language and literacy needs of english learners in science. *Equity & Excellence in Education*, 47(2):230–245, 2014.

[24] K. Brennan, C. Balch, and M. Chung. Creative computing. *Harvard Graduate School of Education*, 2014.

[25] K. Brennan and M. Resnick. New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada*, volume 1, page 25, 2012.

[26] N. W. Brickhouse and J. T. Potter. Young women's scientific identity formation in an urban context. *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching*, 38(8):965–980, 2001.

[27] R. A. T. E. L. BRING. English learners in stem subjects: Transforming classrooms, schools, and lives. 2018.

[28] B. A. Brown and K. Ryoo. Teaching science as a language: A "content-first" approach to science teaching. *Journal of Research in Science Teaching*, 45(5):529–553, 2008.

[29] J. E. Brown and J. Doolittle. A cultural, linguistic, and ecological framework for response to intervention with english language learners. *Teaching Exceptional Children*, 40(5):66–72, 2008.

[30] Q. Burke and Y. B. Kafai. 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*, pages 433–438, 2012.

[31] R. W. Bybee. *Achieving scientific literacy: From purposes to practices.* ERIC, 1997.

[32] D. Cairns and S. Areepattamannil. Exploring the relations of inquiry-based teaching to science achievement and dispositions in 54 countries. *Research in science education*, 49(1):1–23, 2019.

[33] A. Calabrese Barton and C. Berchini. Becoming an insider: Teaching science in urban settings. *Theory Into Practice*, 52(1):21–27, 2013.

[34] A. Calabrese Barton, H. Kang, E. Tan, T. B. O'Neill, J. Bautista-Guerra, and C. Brecklin. Crafting a future in science: Tracing middle school girls' identity work over time and space. *American educational research journal*, 50(1):37–75, 2013.

[35] H. B. Carlone, J. Haun-Frank, and A. Webb. Assessing equity beyond knowledge-and skills-based outcomes: A comparative ethnography of two fourth-grade reform-based science classrooms. *Journal of Research in Science Teaching*, 48(5):459–485, 2011.

[36] H. B. Carlone, L. D. Huffling, T. Tomasek, T. A. Hegedus, C. E. Matthews, M. H. Allen, and M. C. Ash. 'unthinkable'selves: Identity boundary work in a summer field ecology enrichment program for diverse youth. *International Journal of Science Education*, 37(10):1524–1546, 2015.

[37] H. B. Carlone, C. M. Scott, and C. Lowder. Becoming (less) scientific: A longitudinal study of students' identity work from elementary to middle school science. *Journal of Research in Science Teaching*, 51(7):836–869, 2014.

[38] V. Chávez and E. Soep. Youth radio and the pedagogy of collegiality. *Harvard Educational Review*, 75(4):409–434, 2005.

[39] K. Ciechanowski, S. Bottoms, A. L. Fonseca, and T. St Clair. Should rey mysterio drink gatorade? cultural competence in afterschool stem programming. *Afterschool Matters*, 21:29–37, 2015.

[40] . E. A. Code.org, CSTA. 2020 State of computer science education: Illuminating disparities, 2020.

[41] D. Cole and A. Espinoza. Examining the academic success of latino students in science technology engineering and mathematics (stem) majors. *Journal of College Student Development*, 49(4):285–300, 2008.

[42] V. Cook. Multicompetence. *The encyclopedia of applied linguistics*, 2012.

[43] J. E. Cote and C. G. Levine. Identity formation. *Agency and Cultural*, 2002.

[44] N. R. Council et al. *National science education standards*. National Academies Press, 1996.

[45] N. R. Council et al. *A framework for K-12 science education: Practices, crosscutting concepts, and core ideas*. National Academies Press, 2012.

[46] N. R. Council et al. Status, prospects, and an agenda for research. *The National Academies of Science*, 2014.

[47] B. A. Crawford. Learning to teach science as inquiry in the rough and tumble of practice. *Journal of research in science teaching*, 44(4):613–642, 2007.

[48] J. W. Creswell and V. L. P. Clark. *Designing and conducting mixed methods research*. Sage publications, 2017.

[49] J. W. Creswell and J. D. Creswell. *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications, 2017.

[50] A. Crump. Introducing langcrit: Critical language and race theory. *Critical Inquiry in Language Studies*, 11(3):207–224, 2014.

[51] S. F. C. N. A. CSforAll.

[52] L. Darling-Hammond. Teacher learning that supports student learning. *Teaching for intelligence*, 2(1):91–100, 2008.

[53] R. Darvin and B. Norton. Identity and a model of investment in applied linguistics. *Annual review of applied linguistics*, 35:36–56, 2015.

[54] C. S. de Souza, A. C. B. Garcia, C. Slaviero, H. Pinto, and A. Repenning. Semiotic traces of computational thinking acquisition. In *International Symposium on End User Development*, pages 155–170. Springer, 2011.

[55] D. Dean Jr and D. Kuhn. Direct instruction vs. discovery: The long view. *Science Education*, 91(3):384–397, 2007.

[56] J. Denner and R. M.A. Engaging families in stem education for young children. 2020. Presented at the Hawaiian International Conference on Education, Honolulu, HI.

[57] N. K. Denzin. The seventh moment: Qualitative inquiry and the practices of a more radical consumer research. *Journal of Consumer Research*, 28(2):324–330, 2001.

[58] J. Dewey. How we think. a restatement of the relation of reflective thinking to the educative process, boston etc.(dc heath and company) 1933. 1933.

[59] A. A. DiSessa. *Changing minds*. MIT press, 2000.

[60] A. A. DiSessa. *Changing minds: Computers, learning, and literacy*. Mit Press, 2001.

[61] L. M. Dorner, M. F. Orellana, and C. P. Li-Grining. "i helped my mom," and it helped me: Translating the skills of language brokers into improved standardized test scores. *American Journal of Education*, 113(3):451–478, 2007.

[62] Z. Dornyei and I. Ottó. Motivation in action: A process model of l2 motivation. 1998.

[63] R. Dorph, P. Shields, J. Tiffany-Morales, A. Hartry, and T. McCaffrey. High hopes–few opportunities: The status of elementary science education in california. strengthening science education in california. *Center for the future of teaching and learning at WestEd*, 2011.

[64] R. A. Duschl, H. A. Schweingruber, and A. W. Shouse. Taking science to school: Learning and teaching science in grades k-8. *Eurasia Journal of Mathematics, Science & Technology Education*, 3(2):163–166, 2007.

[65] J. S. Eccles and A. Wigfield. Schooling's influences on motivation and achievement. *Securing the future: Investing in children from birth to college*, pages 153–181, 2000.

[66] J. Echevarria, C. Richards-Tutor, R. Canges, and D. Francis. Using the siop model to promote the acquisition of language and science concepts with english learners. *Bilingual Research Journal*, 34(3):334–351, 2011.

[67] R. Eglash. *Appropriating technology: Vernacular science and social power*. U of Minnesota Press, 2004.

[68] G. Estrella, J. Au, S. M. Jaeggi, and P. Collins. Is inquiry science instruction effective for english language learners? a meta-analytic review. *AERA open*, 4(2):2332858418767402, 2018.

[69] N. W. Feinstein, S. Allen, and E. Jenkins. Outside the pipeline: Reimagining science education for nonscientists. *Science*, 340(6130):314–317, 2013.

[70] G. F. G. (Firm). Images of computer science: Perceptions among students, parents and educators in the us. 2015.

[71] N. Flores and J. Rosa. Undoing appropriateness: Raciolinguistic ideologies and language diversity in education. *Harvard Educational Review*, 85(2):149–171, 2015.

[72] A. M. Gadermann, M. Guhn, and B. D. Zumbo. Estimating ordinal reliability for likert-type and ordinal item response data: A conceptual, empirical, and practical guide. *Practical Assessment, Research, and Evaluation*, 17(1):3, 2012.

[73] O. García. Bilingualism and translanguaging. *Bilingual education in the 21st century: A global perspective*, pages 42–72, 2009.

[74] O. García and L. Wei. Language, bilingualism and education. In *Translanguaging: Language, bilingualism and education*, pages 46–62. Springer, 2014.

[75] P. Gibbons. Mediating academic language learning through classroom discourse. In *International handbook of English language teaching*, pages 701–718. Springer, 2007.

[76] S. K. Gilmartin, E. Li, and P. Aschbacher. The relationship between interest in physical science/engineering, science class experiences, and family contexts: Variations by gender and race/ethnicity among secondary students. *Journal of Women and Minorities in Science and Engineering*, 12(2-3), 2006.

[77] N. González, L. C. Moll, and C. Amanti. *Funds of knowledge: Theorizing practices in households, communities, and classrooms*. Routledge, 2006.

[78] J. Goode and J. Margolis. Exploring Computer Science: A Case Study of School Reform. *Trans. Comput. Educ.*, 11(2):12:1–12:16, 2011.

[79] J. Goode and J. Margolis. Exploring computer science: A case study of school reform. *ACM Transactions on Computing Education (TOCE)*, 11(2):1–16, 2011.

[80] J. Goode, J. Margolis, and G. Chapman. Curriculum is not enough: The educational theory and research foundation of the exploring computer science professional development model. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pages 493–498, 2014.

[81] S. Graham and C. Latulipe. Cs girls rock: sparking interest in computer science and debunking the stereotypes. In *Proceedings of the 34th SIGCSE technical symposium on Computer science education*, pages 322–326, 2003.

[82] S. E. Grapin, L. Llosa, A. Haas, and O. Lee. Rethinking instructional strategies with english learners in the content areas. *TESOL Journal*, 12(2):e557, 2021.

[83] K. D. Gutiérrez and A. S. Jurow. Social design experiments: Toward equity by design. *Journal of the Learning Sciences*, 25(4):565–598, 2016.

[84] K. D. Gutiérrez and B. Rogoff. Cultural ways of learning: Individual traits or repertoires of practice. *Educational researcher*, 32(5):19–25, 2003.

[85] A. Haas, S. E. Grapin, D. Wendel, L. Llosa, and O. Lee. How fifth-grade english learners engage in systems thinking using computational models. *Systems*, 8(4):47, 2020.

[86] M. A. K. Halliday. Explorations in the functions of language. 1973.

[87] D. B. Harlow. The excitement and wonder of teaching science: What pre-service teachers learn from facilitating family science night centers. *Journal of Science Teacher Education*, 23(2):199–220, 2012.

[88] P. Häussler and L. Hoffmann. An intervention study to enhance girls' interest, self-concept, and achievement in physics classes. *Journal of research in science teaching*, 39(9):870–888, 2002.

[89] S. Hidi. Interest and its contribution as a mental resource for learning. *Review of Educational research*, 60(4):549–571, 1990.

[90] J. Howell, C. Tofel-Grehl, D. Fields, and G. Ducamp. E-textiles to teach electricity: An experiential, aesthetic, handcrafted approach to science. *Teacher pioneers: visions from the edge of the map. ETC. Press, Pittsburgh*, pages 232–245, 2016.

[91] H.-F. Hsieh and S. E. Shannon. Three approaches to qualitative content analysis. *Qualitative health research*, 15(9):1277–1288, 2005.

[92] N. V. Ivankova, J. W. Creswell, and S. L. Stick. Using mixed-methods sequential explanatory design: From theory to practice. *Field methods*, 18(1):3–20, 2006.

[93] S. Jacob, L. Garcia, and M. Warschauer. Leveraging multilingual identities in computer science education. In *Technology and the Psychology of Second Language Learners and Users*, pages 309–331. Springer, 2020.

[94] S. Jacob, H. Nguyen, L. Garcia, D. Richardson, and M. Warschauer. Teaching computational thinking to multilingual students through inquiry-based learning. In *2020 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, volume 1, pages 1–8. IEEE, 2020.

[95] S. Jacob, H. Nguyen, C. Tofel-Grehl, D. Richardson, and M. Warschauer. Teaching computational thinking to english learners. *NYS TESOL journal*, 5(2), 2018.

[96] S. R. Jacob and M. Warschauer. Computational thinking and literacy. *Journal of Computer Science Integration*, 1(1), 2018.

[97] B. James DiSalvo, S. Yardi, M. Guzdial, T. McKlin, C. Meadows, K. Perry, and A. Bruckman. African american men constructing computing identity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2967–2970, 2011.

[98] J. Janzen. Teaching english language learners in the content areas. *Review of Educational Research*, 78(4):1010–1038, 2008.

[99] Y. Kafai, C. Proctor, and D. Lui. 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, 2020.

[100] Y. Kafai, K. Searle, C. Martinez, and B. Brayboy. Ethnocomputing with electronic textiles: Culturally responsive open design to broaden participation in computing in american indian youth and communities. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pages 241–246, 2014.

[101] Y. B. Kafai and Q. Burke. *Connected code: Why children need to learn programming*. Mit Press, 2014.

[102] H. Kang, A. Calabrese Barton, E. Tan, S. D Simpkins, H.-y. Rhee, and C. Turner. How do middle school girls of color develop stem identities? middle school girls' participation in science activities and identification with stem careers. *Science Education*, 103(2):418–439, 2019.

[103] S. Katz, J. Aronis, D. Allbritton, C. Wilson, and M. L. Soffa. Gender and race in predicting achievement in computer science. *IEEE Technology and Society Magazine*, 22(3):20–27, 2003.

[104] A. Keselman. Supporting inquiry learning by promoting normative understanding of multivariable causality. *Journal of Research in Science Teaching*, 40(9):898–921, 2003.

[105] P. A. Kirschner and J. Sweller. Richard e. clark,"why minimal guidance during instruction does not work: an analysis of the failure of constructivist, discovery, problem-based, experiential and inquiry-based teaching". *Educational Psychologist*, 41:2, 2006.

[106] D. Klahr and M. Nigam. The equivalence of learning paths in early science instruction: Effects of direct instruction and discovery learning. *Psychological science*, 15(10):661–667, 2004.

[107] H. Kohl. *I won't learn from you*. New Press, 1995.

[108] P. Kruit, R. Oostdam, E. Van den Berg, and J. Schuitema. Effects of explicit instruction on the acquisition of students' science inquiry skills in grades 5 and 6 of primary education. *International Journal of Science Education*, 40(4):421–441, 2018.

[109] R. A. Krystyniak and H. W. Heikkinen. Analysis of verbal interactions during an extended, open-inquiry general chemistry laboratory investigation. *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching*, 44(8):1160–1186, 2007.

[110] R. E. Ladner and M. Israel. For all" in" computer science for all. *Communications of the ACM*, 59(9):26–28, 2016.

[111] J. Lave and E. Wenger. *Situated learning: Legitimate peripheral participation*. Cambridge university press, 1991.

[112] M. D. LeCompte, J. J. Schensul, B. K. Nastasi, S. P. Borgatti, et al. *Enhanced ethnographic methods: Audiovisual techniques, focused group interviews, and elicitation*. Rowman Altamira, 1999.

[113] I. Lee, F. Martin, J. Denner, B. Coulter, W. Allan, J. Erickson, J. Malyn-Smith, and L. Werner. Computational thinking for youth in practice. *Acm Inroads*, 2(1):32–37, 2011.

[114] J.-S. Lee and N. K. Bowen. Parent involvement, cultural capital, and the achievement gap among elementary school children. *American educational research journal*, 43(2):193–218, 2006.

[115] O. Lee and S. H. Fradd. Science for all, including students from non-english-language backgrounds. *Educational researcher*, 27(4):12–21, 1998.

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

[117] O. Lee, L. Llosa, F. Jiang, A. Haas, C. O'Connor, and C. D. Van Booven. Elementary teachers' science knowledge and instructional practices: Impact of an intervention focused on english language learners. *Journal of Research in Science Teaching*, 53(4):579–597, 2016.

[118] O. Lee and A. Stephens. English learners in stem subjects: Contemporary views on stem subjects and language with english learners. *Educational researcher*, 49(6):426–432, 2020.

[119] C. Linse. English language learner a term that warrants scrutiny. *Journal of Educational Thought/Revue de la Pensée Educative*, pages 107–122, 2013.

[120] S. Loucks-Horsley, N. Love, K. Stiles, and S. Mundry. Hewson. pw (2003). *Designing professional development for teachers of science and mathematics*, 2, 2010.

[121] E. Lyon, S. Tolbert, J. Solis, T. Stoddart, I. Salinas, C. Knox, B. Roth, and M. Butler. Teaching english learners through science-language integration: Linking a conceptual framework to secondary teacher preparation. In *Strand 7 Symposium at the annual conference of the National Association of Research in Science Teaching, Baltimore, MD*, 2016.

[122] E. G. Lyon, S. Tolbert, J. Solís, P. Stoddart, and G. C. Bunch. *Secondary science teaching for English learners: Developing supportive and responsive learning contexts for sense-making and language development*. Rowman & Littlefield, 2016.

[123] S. L. Manz. A strategy for previewing textbooks: teaching readers to become thieves.(teaching ideas). *The Reading Teacher*, 55(5):434–436, 2002.

[124] J. Margolis, J. J. Ryoo, C. D. Sandoval, C. Lee, J. Goode, and G. Chapman. Beyond access: Broadening participation in high school computer science. *ACM Inroads*, 3(4):72–78, 2012.

[125] A. Martin, F. McAlear, and A. Scott. Path not found: Disparities in access to computer science courses in california high schools. *Online Submission*, 2015.

[126] J. McFarland, B. Hussar, J. Zhang, X. Wang, K. Wang, S. Hein, M. Diliberti, E. F. Cataldi, F. B. Mann, and A. Barmer. The condition of education 2019. nces 2019-144. *National Center for Education Statistics*, 2019.

[127] J. L. Meece, P. Herman, and B. L. McCombs. Relations of learner-centered teaching practices to adolescents' achievement goals. *International Journal of Educational Research*, 39(4-5):457–475, 2003.

[128] K. Menken. Emergent bilingual students in secondary school: Along the academic language and literacy continuum. *Language Teaching*, 46(4):438, 2013.

[129] S. Michaels and C. O'Connor. Conceptualizing talk moves as tools: Professional development approaches for academically productive discussion. *Socializing intelligence through talk and dialogue*, pages 347–362, 2015.

[130] P. Mishra, A. Yadav, D.-P. R. Group, et al. Rethinking technology & creativity in the 21st century. *TechTrends*, 57(3):10–14, 2013.

[131] E. B. Moje, K. M. Ciechanowski, K. Kramer, L. Ellis, R. Carrillo, and T. Collazo. Working toward third space in content area literacy: An examination of everyday funds of knowledge and discourse. *Reading research quarterly*, 39(1):38–70, 2004.

[132] L. C. Moll, C. Amanti, D. Neff, and N. Gonzalez. Funds of knowledge for teaching: Using a qualitative approach to connect homes and classrooms. *Theory into practice*, 31(2):132–141, 1992.

[133] J. M. Morse and J. L. Bottorff. The emotional experience of breast expression. *Journal of Nurse-Midwifery*, 33(4):165–170, 1988.

[134] R. P. Moses and C. Cobb. Organizing algebra: The need to voice a demand. *Social Policy*, 31(4):4–12, 2001.

[135] S. Motha. Is an antiracist and decolonizing applied linguistics possible? connecting the past to the future. In *2020 conference of the American Association for Applied Linguistics (AAAL)*. AAAL, 2020.

[136] N. S. Nasir and J. Cooks. Becoming a hurdler: How learning settings afford identities. *Anthropology & Education Quarterly*, 40(1):41–61, 2009.

[137] N. S. Nasir and V. Hand. From the court to the classroom: Opportunities for engagement, learning, and identity in basketball and classroom mathematics. *The Journal of the Learning Sciences*, 17(2):143–179, 2008.

[138] H. Nguyen, L. Garcia, S. Jacob, D. Richardson, and M. Warschauer. Reflection as formative assessment of computational thinking in elementary grades. 2020.

[139] B. Norton. Non-participation, imagined communities and the language classroom. *Learner contributions to language learning: New directions in research*, 6(2):159–171, 2001.

[140] B. A. Nosek, M. R. Banaji, and A. G. Greenwald. Math= male, me= female, therefore math≠ me. *Journal of personality and social psychology*, 83(1):44, 2002.

[141] G. Nugent, B. Barker, G. Welch, N. Grandgenett, C. Wu, and C. Nelson. A model of factors contributing to stem learning and career orientation. *International Journal of Science Education*, 37(7):1067–1088, 2015.

[142] B. of Labor Statistics. Occupational outlook handbook: Computer and information technology, 2018.

[143] P. Ordóñez Franco, J. Carroll-Miranda, M. López Delgado, E. Gerena López, and G. Rodríguez Gómez. Incorporating computational thinking in the classrooms of puerto rico: How a mooc served as an outreach and recruitment tool for computer science education. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, pages 296–301, 2018.

[144] J. Osborne. Attitudes towards science: An update. In *A paper presented at the Annual Meeting of the American Educational Research Association, San Diego, California, April, 2009*, pages 13–17, 2009.

[145] J. Osborne, S. Collins, M. Ratcliffe, R. Millar, and R. Duschl. What "ideas-about-science" should be taught in school science? a delphi study of the expert community. *Journal of research in science teaching*, 40(7):692–720, 2003.

[146] D. Oyserman, D. Brickman, and M. Rhodes. School success, possible selves, and parent school involvement. *Family Relations*, 56(5):479–489, 2007.

[147] D. Paris. Culturally sustaining pedagogy: A needed change in stance, terminology, and practice. *Educational researcher*, 41(3):93–97, 2012.

[148] D. Paris and H. S. Alim. What are we seeking to sustain through culturally sustaining pedagogy? a loving critique forward. *Harvard educational review*, 84(1):85–100, 2014.

[149] M. Pedaste, M. Mäeots, L. A. Siiman, T. De Jong, S. A. Van Riesen, E. T. Kamp, C. C. Manoli, Z. C. Zacharia, and E. Tsourlidaki. Phases of inquiry-based learning: Definitions and the inquiry cycle. *Educational research review*, 14:47–61, 2015.

[150] M. Pedaste and T. Sarapuu. Developing an effective support system for inquiry learning in a web-based environment. *Journal of computer assisted learning*, 22(1):47–62, 2006.

[151] W. R. Penuel and B. J. Fishman. Large-scale science education intervention research we can use. *Journal of research in science teaching*, 49(3):281–304, 2012.

[152] W. R. Penuel, T. Lee, and B. Bevan. Designing and building infrastructures to support equitable stem learning across settings. *Research+ Practice Collaboratory Research Synthesis. San Francisco: The exploratorium*, 2014.

[153] K. A. Peppler. Media arts: Arts education for a digital age. *Teachers College Record*, 112(8):2118–2153, 2010.

[154] K. A. Peppler and Y. B. Kafai. From supergoo to scratch: Exploring creative digital media production in informal learning. *Learning, media and technology*, 32(2):149–166, 2007.

[155] K. A. Peppler and M. Warschauer. 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, 2011.

[156] N. Pinkard, S. Erete, C. K. Martin, and M. McKinney de Royston. 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, 2017.

[157] N. Planas and M. Setati. Bilingual students using their languages in the learning of mathematics. *Mathematics Education Research Journal*, 21(3):36–59, 2009.

[158] J. L. Polman, A. Newman, E. W. Saul, and C. Farrar. Adapting practices of science journalism to foster science literacy. *Science Education*, 98(5):766–791, 2014.

[159] C. Quintana, M. Zhang, and J. Krajcik. A framework for supporting metacognitive aspects of online inquiry through software-based scaffolding. *Educational Psychologist*, 40(4):235–244, 2005.

[160] B. J. Reiser. Scaffolding complex learning: The mechanisms of structuring and problematizing student work. *The Journal of the Learning sciences*, 13(3):273–304, 2004.

[161] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, et al. Scratch: programming for all. *Communications of the ACM*, 52(11):60–67, 2009.

[162] K. M. Rich, A. Yadav, and C. V. Schwarz. Computational thinking, mathematics, and science: Elementary teachers' perspectives on integration. *Journal of Technology and Teacher Education*, 27(2):165–205, 2019.

[163] S. Rodriguez, A. Pilcher, and N. Garcia-Tellez. The influence of familismo on latina student stem identity development. *Journal of Latinos and Education*, 20(2):177–189, 2021.

[164] B. Rogoff. Evaluating development in the process of participation: Theory, methods, and practice building on each other. *Change and development: Issues of theory, method, and application*, pages 265–285, 1997.

[165] D. H. Rose and A. Meyer. *Teaching every student in the digital age: Universal design for learning.* ERIC, 2002.

[166] A. S. Rosebery. *Teaching science to English language learners: Building on students' strengths.* NSTA Press, 2008.

[167] A. S. Rosebery, B. Warren, and F. R. Conant. Appropriating scientific discourse: Findings from language minority classrooms. *The Journal of the Learning Sciences*, 2(1):61–94, 1992.

[168] D. Royal and A. Swift. Us minority students less exposed to computer science. *Gallup, October*, 2016.

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

[170] J. J. Ryoo, J. Margolis, C. Estrada, T. C. Tanksley, D. Guest-Johnson, and S. Mendoza. Student voices: Equity, identity, and agency in cs classrooms. In *2019 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, pages 1–2. IEEE, 2019.

[171] J. Salac, C. Thomas, C. Butler, A. Sanchez, and D. Franklin. Tipp&see: A learning strategy to guide students through use-modify scratch activities. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, pages 79–85, 2020.

[172] J. Saldaña. *The coding manual for qualitative researchers.* sage, 2021.

[173] K. Schildkamp and W. Kuiper. Data-informed curriculum reform: Which data, what purposes, and promoting and hindering factors. *Teaching and teacher education*, 26(3):482–496, 2010.

[174] D. A. Schön. *The reflective practitioner: How professionals think in action.* Routledge, 2017.

[175] K. A. Searle and Y. B. Kafai. Boys' needlework: Understanding gendered and indigenous perspectives on computing and crafting with electronic textiles. In *ICER*, pages 31–39, 2015.

[176] S. F. Sembiante and Z. Tian. Culturally sustaining approaches to academic languaging through systemic functional linguistics, 2021.

[177] P. Sengupta, A. Dickes, and A. Farris. Toward a phenomenology of computational thinking in stem education. *Computational thinking in the STEM disciplines*, pages 49–72, 2018.

[178] M. Shaw and Y. Kafai. Charting the identity turn in k-12 computer science education: Developing more inclusive learning pathways for identities. 2020.

[179] L. M. Shea, J. H. Sandholtz, and T. B. Shanahan. We are all talking: A whole-school approach to professional development for teachers of english learners. *Professional Development in Education*, 44(2):190–208, 2018.

[180] L. M. Shea and T. B. Shanahan. We are all talking to learn science: Finding the right fit.

[181] C. Shettle, S. Roey, J. Mordica, R. Perkins, C. Nord, J. Teodorovic, M. Lyons, C. Averett, D. Kastberg, and J. Brown. The nation's report card [tm]: America's high school graduates. nces 2007-467. *National Center for Education Statistics*, 2007.

[182] J. Siegmund, C. Kästner, S. Apel, C. Parnin, A. Bethmann, T. Leich, G. Saake, and A. Brechmann. Understanding understanding source code with functional magnetic resonance imaging. In *Proceedings of the 36th international conference on software engineering*, pages 378–389, 2014.

[183] S. D. Simpkins, J. A. Fredricks, and J. S. Eccles. Families, schools, and developing achievement-related motivations and engagement. 2015.

[184] S. D. Simpkins, C. D. Price, and K. Garcia. Parental support and high school students' motivation in biology, chemistry, and physics: Understanding differences among latino and caucasian boys and girls. *Journal of Research in Science Teaching*, 52(10):1386–1407, 2015.

[185] M. Smith. Computer science for all. 2016. *The White House President Barack Obama: USA*, 2016.

[186] J. P. Smith III, A. A. DiSessa, and J. Roschelle. Misconceptions reconceived: A constructivist analysis of knowledge in transition. *The journal of the learning sciences*, 3(2):115–163, 1994.

[187] C. M. Steele. Race and the schooling of black americans. *The Atlantic Monthly*, 269(4):68–78, 1992.

[188] C. M. Steele. A threat in the air: How stereotypes shape intellectual identity and performance. *American psychologist*, 52(6):613, 1997.

[189] J. Stockard, T. W. Wood, C. Coughlin, and C. Rasplica Khoury. The effectiveness of direct instruction curricula: A meta-analysis of a half century of research. *Review of Educational Research*, 88(4):479–507, 2018.

[190] J. Sullivan and M. Hatton. Math and science night. *Science and Children*, 48(5):58–63, 2011.

[191] R. H. Tai, C. Q. Liu, A. V. Maltese, and X. Fan. Planning early for careers in science. *Life sci*, 1(0.2), 2006.

[192] M. Tissenbaum, J. Sheldon, and H. Abelson. From computational thinking to computational action. *Communications of the ACM*, 62(3):34–36, 2019.

[193] C. Tofel-Grehl, D. Fields, K. Searle, C. Maahs-Fladung, D. Feldon, G. Gu, and C. Sun. Electrifying engagement in middle school science class: Improving student interest through e-textiles. *Journal of Science Education and Technology*, 26(4):406–417, 2017.

[194] N. Trautmann, J. MaKinster, and L. Avery. What makes inquiry so hard?(and why is it worth it?). In *annual meeting of the national association for research in science teaching, Vancouver, BC, Canada*, 2004.

[195] R. Tukachinsky, D. Mastro, and M. Yarchi. The effect of prime time television ethnic/racial stereotypes on latino and black americans: A longitudinal national level study. *Journal of Broadcasting & Electronic Media*, 61(3):538–556, 2017.

[196] S. Vakil. Ethics, Identity, and Political Vision: Toward a Justice-Centered Approach to Equity in Computer Science Education. *Harvard Educational Review*, 88(1):26–52, 3 2018.

[197] K. Van Horne and P. Bell. Youth disciplinary identification during participation in contemporary project-based science investigations in school. *Journal of the Learning Sciences*, 26(3):437–476, 2017.

[198] S. Vogel, C. Hoadley, L. Ascenzi-Moreno, and K. Menken. The role of translanguaging in computational literacies: Documenting middle school bilinguals' practices in computer science integrated units. In *Proceedings of the 50th ACM technical symposium on computer science education*, pages 1164–1170, 2019.

[199] S. Vogel, C. Hoadley, A. R. Castillo, and L. Ascenzi-Moreno. 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, 2020.

[200] S. Vogel, R. Santo, and D. Ching. 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*, pages 609–614, 2017.

[201] A. Wagh, K. Cook-Whitt, and U. Wilensky. Bridging inquiry-based science and constructionism: Exploring the alignment between students tinkering with code of computational models and goals of inquiry. *Journal of Research in Science Teaching*, 54(5):615–641, 2017.

[202] J. Wang, H. Hong, J. Ravitz, and S. Hejazi Moghadam. Landscape of k-12 computer science education in the us: Perceptions, access, and barriers. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, pages 645–650, 2016.

[203] M. Warschauer. *Technology and social inclusion: Rethinking the digital divide*. MIT press, 2004.

[204] D. Weintrop, E. Beheshti, M. Horn, K. Orton, K. Jona, L. Trouille, and U. Wilensky. Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1):127–147, 2016.

[205] L. Werner, J. Denner, S. Campe, and D. C. Kawamoto. The fairy performance assessment: Measuring computational thinking in middle school. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, pages 215–220, 2012.

[206] A. Wigfield. Expectancy-value theory of achievement motivation: A developmental perspective. *Educational psychology review*, 6(1):49–78, 1994.

[207] M. Windschitl. Inquiry projects in science teacher education: What can investigative experiences reveal about teacher thinking and eventual classroom practice? *Science education*, 87(1):112–143, 2003.

[208] J. M. Wing. Computational Thinking. *Communications of the ACM*, 49(3):33–35, 2005.

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

[210] A. Wright, M. A. Gottfried, and V.-N. Le. A kindergarten teacher like me: The role of student-teacher race in social-emotional development. *American Educational Research Journal*, 54(1_suppl):78S–101S, 2017.

[211] W. E. Wright. The effects of high stakes testing in an inner-city elementary school: The curriculum, the teachers, and the english language learners. *Current Issues in Education*, 5, 2002.

[212] N. Zarrett and J. Eccles. The role of the family and community in extracurricular activity. *AERA Monograph series: promising practices for family and community involvement during high school*, 4:27–51, 2009.

[213] L. Zhang. Withholding answers during hands-on scientific investigations? comparing effects on developing students' scientific knowledge, reasoning, and application. *International Journal of Science Education*, 40(4):459–469, 2018.

[214] N. Zhou, Y. Cao, S. Jacob, and D. Richardson. Teacher perceptions of equity in high school computer science classrooms. *ACM Transactions on Computing Education (TOCE)*, 20(3):1–27, 2020.

[215] S. G. Zwiep and W. J. Straits. Inquiry science: The gateway to english language proficiency. *Journal of Science Teacher Education*, 24(8):1315–1331, 2013.

[216] S. G. Zwiep and W. J. Straits. The integration of english language development and inquiry science into a blended professional development design. In *Science Teacher Preparation in Content-Based Second Language Acquisition*, pages 163–177. Springer, 2017.