A Qualitative Examination of Second Grade Teachers' Experiences and Attitudes Around Coding and Robotics Education

A dissertation submitted by

Madhumita Govindarajan (Madhu Govind)

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Child Study and Human Development

Tufts University

May 2022

Committee Members

Marina Umaschi Bers, Ph.D. (Chair)

Eliot-Pearson Department of Child Study and Human Development, Tufts University

Christine McWayne, Ph.D.

Eliot-Pearson Department of Child Study and Human Development, Tufts University

X. Christine Wang, Ph.D.

Department of Learning and Instruction, State University of New York at Buffalo

Rebecca Dore, Ph.D.

The Crane Center for Early Childhood Research and Policy, The Ohio State University

Abstract

In recent years there has been an increased push for K-12 computer science education. A major reason for this push is the growing evidence that introducing young children to coding can enhance their interest and promote their learning of foundational skills needed to thrive in today's technologically rich world. However, little research has focused on early elementary teachers and their diverse experiences and attitudes around teaching coding and robotics. This dissertation addresses this gap by examining a sample of second grade educators (*N* = 15) from six elementary schools in a large U.S. public school district. The schools were selected to pilot the Coding as Another Language (CAL) – KIBO program, which consisted of a training and curriculum that emphasized the pedagogical overlaps between computer science and literacy using the KIBO educational robotics platform. Teachers attended a full-day professional development training and subsequently implemented the CAL-KIBO curriculum in their classrooms. Teachers were interviewed at various points before, during, and after the training and curriculum implementation. Analysis of teacher interviews revealed five major categories encapsulating their experiences and attitudes: 1) knowledge of coding and robotics, and of teaching, 2) positive and negative attitudes about KIBO, teaching coding and robotics, and integrating coding with other content areas, 3) beliefs about coding, inclusion, and overall principles of education, 4) use and perception of human and material supports, and 5) strengths and challenges of instructional behaviors and pedagogical strategies. From these categories, three groups of teachers were identified, each of which shared similar experiences and attitudes. The first group of teachers, all with over ten years of teaching experience, voiced substantial challenges with lesson implementation and expressed predominantly negative attitudes towards teaching coding and robotics, as well as negative perceptions of support. The second group, with an intermediate level of teaching experience, also expressed challenges with lesson implementation but exhibited more moderate attitudes and perceptions of support. The third group, who had a mixed range of teaching experience, exhibited positive attitudes, perceptions of support, and beliefs about coding, in addition to sharing many highlights of their overall program experience. Case study examples illustrate teachers' varied experiences and attitudes. Implications of these findings, study limitations, and recommendations for research, practice, and policy are presented.

Keywords: early elementary, coding, qualitative study, teachers, thematic analysis, robotics

Acknowledgements

This dissertation would not have been possible without the support of many, many people for whom I am forever grateful. First and foremost, I would like to thank the educators, students, and administrators at Norfolk Public Schools. Thank you for your time and willingness to go above and beyond your regular duties to engage in new learning experiences and participate in this research. I am especially grateful to Angela de Mik for her unwavering support and dedication. Your uplifting spirit and enthusiasm for this work are truly what made this project successful.

To my mentor, advisor, and committee chair Marina Umaschi Bers, there are not enough words to express my gratitude. Thank you for taking a chance on me and pushing me continuously to believe in myself. I am immensely grateful for all the opportunities you have provided, opportunities I never even dreamed were possible. You have shown me what it means to be a hard-working researcher, a passionate teacher, a loving family member, and above all, a good person. Your strong example is something I will always continue to strive towards and aim to pass forth to others.

I am also incredibly fortunate to have the strong mentorship and support of an amazing dissertation committee: Christine McWayne, X. Christine Wang, and Rebecca Dore. Each of you provided invaluable feedback and expert guidance I needed to continue in the right direction and successfully complete my dissertation. At each phase of writing and presenting my ideas, your thoughtful questions challenged me to sharpen my thinking and ultimately made this work richer and more meaningful. Thank you so much for your time and support.

I am deeply grateful for my DevTech family who made my doctoral program experience unforgettable. Thank you to Amanda Strawhacker and Emily Relkin for welcoming me with open arms in my first year, teaching a southern gal like myself how to survive a Boston winter, and showing me the DevTech ropes. I have learned so much from each of you, and your respective research contributions have inspired me in so many ways. Thank you to Ziva Hassenfeld, Laura de Ruiter, Amanda Sullivan, and Angie Kalthoff for your mentorship during the most uncertain and challenging years of graduate school. I am a better researcher and educator because of you. Thank you to Jess Blake-West, Riva Dhamala, Anne Drescher, and Melissa Viezel for always lending a helping hand, even when you have a lot on your own plate. Thank you to Alexa Hasse for helping me code my dissertation data, and to Aim Unahalekhaka and Tess Levinson for their feedback and support throughout my writing process. I also want to thank the following undergraduate and graduate students who worked on the Norfolk project: Maya Morris, Jaclyn Tsiang, Ari Lerner, Brendan Brennan, Hasan Khan, Patrick Nero, Megan Bennie, Rachel Viselman, Kaylyn Adams, and Hannah Riehl. Each of you made significant contributions to this work, and I could not have gotten to this point without you.

Thank you to Barbara and Brad Evans for their generous funding that supported my doctoral research. I hope one day to be able to pass your generosity forward to young aspiring scholars. I would also like to extend my gratitude to the entire Eliot-Pearson community of professors, staff, and fellow students who have supported, challenged, and inspired me. Despite not being physically in the building for two years due to the COVID-19 pandemic, I know the memories I have made in my time here will be something I cherish forever. Coming to Eliot-Pearson and pursuing a Ph.D. would have never been possible without the amazing teachers and students I have encountered throughout my educational journey. Thank you to my PreK-12 teachers for instilling a love of learning in me; my extracurricular teachers for providing opportunities to express myself creatively and artistically; my college professors for introducing me to child development as a field of study; and (when I was a teacher myself) my middle school students for teaching me important life lessons about hard work and perseverance. I am also grateful to the many educators and early childhood professionals I have had the pleasure of working with over these last few years. Your endeavors to improve the lives of children, families, and educators in your respective communities are an inspiration for this work.

To close out my acknowledgements, I extend my humblest gratitude to my family and friends who have witnessed my growth through these years and supported me through the highs and lows. Thank you to my father for sparking my curiosity about computer science at an early age and proudly cheering me on by signing up for our newsletters and sharing articles I would find useful and relevant to my work. Thank you to my mother for always encouraging me to pursue a career that fulfilled me internally and showing me first-hand the kind of positive impact an educator can have on a child’s life. Thank you to my brother and sister-in-law for their support and encouragement over the years. You never cease to amaze me by everything you are and everything you do. Thank you to our family dog Simba for being the greatest, most cuddly writing companion, a true blessing to have while working from home in a global pandemic. Thank you to my family and friends here and in India for supporting me from afar and relieving me of wedding planning responsibilities so I could focus on writing. To my grandmothers, both of whom passed away during my time in the doctoral program, thank you for your everlasting love and blessings.

Finally, I extend my sincerest appreciation for my fiancé Neal, for encouraging me to take the GRE and apply to doctoral programs, being my biggest cheerleader, and supporting and loving me every step of the way. Thank you for our FaceTime work sessions during our years of long distance and weekend getaways spent at coffee shops and libraries. Thank you for all the times I was too exhausted to cook a meal or too stressed to take a break. I am so grateful for you and excited to celebrate our marriage in the very near future. To Neal's parents, sisters, and extended family, I feel so lucky to have such a large family network and extended support system. Thank you for your love and encouragement in this endeavor.

The material in this dissertation is based upon work supported by the U.S. Department of Defense (DoD). Any opinions, findings, and conclusions or recommendations expressed in this article are those of the author and do not necessarily reflect the views of the DoD.

List of Tables

List of Figures

Table of Contents

A Qualitative Examination of Second Grade Teachers' Experiences and Attitudes Around Coding and Robotics Education

Chapter One: Introduction

In the last several decades, teaching computer science (CS) in schools has shifted from a "nice-to-have" to a "must-have". Thirty-nine U.S. states have passed four or more policies to introduce new CS standards, curricular frameworks, and coding technologies in K-12 schools (Code.org, CSTA, & ECEP Alliance, 2021; Stanton et al., 2017). Alongside these large-scale policy advancements, non-profit organizations and funding agencies have devoted unprecedented resources to developing tools and curricula that engage young children in learning to code. For instance, the U.S. Department of Education marked CS as an exclusive "competitive priority" within one of its major grant programs and committed to allocating at least $200 million each year towards STEM (Science, Technology, Engineering and Mathematics) and CS education initiatives (Code.org, 2021).

The rationale for these massive efforts stems from multiple factors. Computing occupations are the primary source of new jobs in the U.S., so there is a growing demand to equip students with the knowledge and skills needed to fulfill this economic need (Fayer, Lacey, & Watson, 2017). Another factor is the current underrepresentation of students from marginalized racial and ethnic groups in the computing field. In response, many early CS and STEM interventions have been developed to get young children interested in coding before gender or racial stereotypes about the field are deeply ingrained (Bers, 2018; Code.org, 2021; Sullivan, 2019). Early interventions have also shown additional benefits for children such as enhanced problem-solving abilities and computational thinking skills, which are a set of underlying cognitive abilities related to but not exclusive to CS (Bers et al., 2021; Phillips & Brooks, 2017; U.S. Department of Education, Office of Educational Technology, 2017; Wing, 2006).

This growing recognition that coding can and should be introduced earlier in children's educational pathways has led to a heightened focus on early elementary educators. Multiple studies have found that a well-prepared and knowledgeable teacher is the most important school-related factor influencing student achievement (Darling-Hammond, 2000; Podolsky et al., 2019; Rice, 2003; Rivkin, Hanushek, & Kain, 2005). However, due to limited professional guidance, time and other barriers, early elementary educators often lack the knowledge and understanding about coding, and about developmentally appropriate pedagogical approaches to integrate coding effectively in their classrooms. Furthermore, teachers have varying attitudes and beliefs about coding and teaching coding to young children, which may influence their ability and interest to engage in coding education. New professional development models and strategies have aimed to address these challenges, but there is still limited research on how early childhood educators experience learning and teaching CS in their classrooms, often for the very first time.

This dissertation addresses this gap by examining a sample of second grade classrooms from a large public school district in Virginia. In February of 2016, Virginia became a national leader in CS education by passing legislation to make coding a mandatory topic in all K-12 Virginia schools (Virginia Department of Education, 2021). In partnership with the DevTech Research Group at Tufts University, the Norfolk Public School (NPS) district sought to pilot professional development and curricular resources for K-2 educators in eight elementary schools in 2018-2020. This dissertation will examine a subset of data collected from this larger NPS study, focusing on $N = 15$ second grade educators and their students from six elementary schools.

Chapter Two: Review of the Literature

This dissertation draws upon theoretical frameworks and ideas from multiple disciplines, including education, child development, computer science, and psychology. The interplay of these ideas forms our collective understanding about teaching coding and robotics in early elementary classrooms. To organize these related but distinct ideas, this chapter is divided into four sections. The first section provides an overview of the literature on teacher knowledge, attitudes, beliefs, and instructional behaviors and how these constructs have been studied in both STEM and non-STEM contexts. The second section provides a deeper look at CS as a disciplinary topic in K-12 education and the global movement to promote not just coding, but also computational thinking, for all students. Building upon this literature, the third section presents a summary of research on coding and robotics initiatives for young children and discusses the design affordances of tangible interfaces such as robotics kits. The fourth and final section synthesizes the existing literature on teaching robotics in elementary classrooms, with a specific focus on teachers' knowledge, attitudes, beliefs, and behaviors.

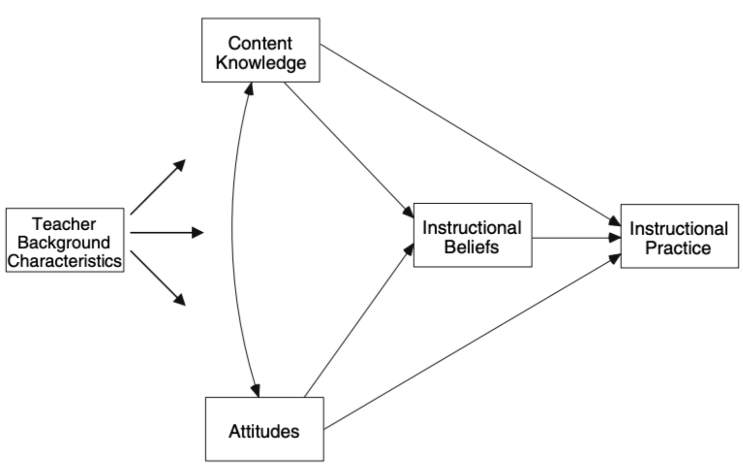
Teacher Knowledge, Attitudes, Beliefs, and Instructional Behaviors

To understand the complex relationships among teacher knowledge, attitudes, beliefs, and instructional behaviors, it is useful to first define these terms. Knowledge is commonly defined as information or skills acquired by a person through experience or education. For instance, one can acquire knowledge about a specific content area or even how to teach. Attitudes refer to feelings towards an object or construct and often hold judgmental value (e.g., liking or disliking something). Beliefs refer to statements that a person considers to be true, regardless of whether the statement is actually accurate. Attitudes and beliefs influence one another; what a person believes to be true about an object (belief) determines how they feel about that object (attitude). A teacher's knowledge, attitudes, and beliefs all play an important role in what the teacher does and how the teacher teaches in their classroom, which together make up the teacher's instructional behaviors or practices.

Ernest (1989) theorized a descriptive model that outlined different types of knowledge, attitudes, and beliefs of a teacher and how these constructs relate to a teacher's instructional practices. This theoretical model (see Figure 1) was originally devised for the specific context of teaching mathematics but is applicable and useful to any content area, including coding. According to Ernest's model, teacher knowledge is comprised of multiple components, including knowledge of (a) the specific subject matter, (b) other subject matter, (c) pedagogy and curriculum, (d) classroom management, (e) teaching context, and (f) education. Teacher beliefs include (a) beliefs regarding the overall nature of the subject matter, (b) models of teaching and learning the subject matter, and (c) principles of education. Teacher attitudes include (a) attitudes towards the subject matter, and (b) attitudes towards teaching the subject matter. Each of these components of teacher knowledge, beliefs, and attitudes are useful to examine in this study.

Figure 1

Relationships Among Teachers' Knowledge, Attitudes, Beliefs, and Instructional Behaviors



Note. Figure reprinted from "The Relationship among Elementary Teachers' Content Knowledge, Attitudes, Beliefs, and Practices" by Wilkins, J. L. M., 2008, *Journal of Mathematics Teacher Education, 11,* p. 145.

As shown in Figure 1, content knowledge, instructional beliefs, and attitudes are all posited to have a direct impact on teachers' instructional practices, and the relationship between knowledge and attitudes is viewed as bidirectional. Having more positive attitudes about a topic may lead to more opportunities to seek out knowledge on that topic. Similarly, acquiring more knowledge about a topic may lead to enhanced attitudes towards that topic. In addition, teachers' beliefs are hypothesized to mediate the effect of teachers' knowledge and attitudes. Ernest's model also accounts for teachers' individual characteristics. Factors such as years of teaching experience, courses taken, and educational level may influence the relationships among teacher knowledge, attitudes, beliefs, and instructional behaviors and might be useful factors to explore in this dissertation.

Ernest's theoretical model parallels models and frameworks presented by other scholars. For instance, Wilson, Shulman and Richert (1987) proposed the following knowledge categories: knowledge of subject matter, pedagogical content knowledge, knowledge of other content, knowledge of curriculum, knowledge of learners, knowledge of educational aims, and general pedagogical knowledge. Shulman's work (1986) on pedagogical content knowledge is a commonly cited framework, particularly in relation to technology integration and Mishra and Koehler's (2006) Technological Pedagogical Content Knowledge framework (TPACK). These frameworks highlight the intersection of knowledge about *what* to teach, *how* to teach, and the *tools* used to teach. In addition, Desimone (2009) proposed a path model for understanding the impact on professional development in shaping teachers' knowledge, attitudes, and beliefs, which thereby shape instructional practices and ultimately student learning. In this dissertation, I draw upon Ernest's theoretical model and these others to conceptualize the various categories of teacher knowledge, attitudes, beliefs, and instructional behaviors within the context of coding and robotics education. However, before diving into this specific context, it is useful to discuss how these constructs have been studied in other STEM and non-STEM domains.

STEM Domains

In addition to Ernest (1989), other scholars have explored the relationships among teacher knowledge, attitudes, beliefs, and practices in mathematics education. Wilkins (2008), for example, found that in a sample of 481 in-service elementary teachers, all three components—knowledge, attitudes, and beliefs—were related to teachers' instructional practices. However, beliefs about mathematics teaching had the strongest effect and partially mediated the effects of knowledge and attitudes on instructional practices. Another example is Gresham and Burleigh (2019), who explored early childhood pre-service teachers ' mathematics anxiety and mathematics efficacy beliefs. Prior research has shown that pre-service teachers who experience mathematics anxiety are more likely to have negative views of mathematics and teach in ways that develop mathematics anxiety in their students. The authors' findings show that a reform-based constructivist teaching method was effective in reducing teachers' mathematics anxiety, which was linked to their efficacy beliefs.

In the context of science education, studies indicate a similar relationship among teacher attitudes, beliefs, and instructional practices. Elementary teachers with more positive attitudes towards science education tend to spend more time teaching science, incorporate hands-on inquiry-based instructional methods, and believe science is an essential subject in elementary students' curriculum (Maier et al., 2013). Bryan and Atwater (2002) argue that teacher beliefs have a significant impact on science teaching and learning, and in particular, beliefs around cultural issues. Their argument parallels what Ernest (1989) referred to when addressing teachers' background characteristics in his model. The broader context of teachers' identities, backgrounds, and experiences plays a key role in shaping their knowledge, attitudes, and beliefs around teaching. Building on this point, van Driel and colleagues (2001) argue that many failed interventions do so because the intervention developers did not consider the teachers, students, and the culture in which the intervention was to be embedded.

Another prominent STEM domain to explore this complex system of teacher variables is computer technology. For example, Hardy (1999) found that higher levels of knowledge about computers were associated with more positive attitudes and less anxiety. Several studies have found linkages among teacher attitudes, skills, and practices around technology integration in classrooms (Alexander et al., 2014; Chen & Chang, 2006; Fenty & McKendry, 2014). However, there is research evidence for teacher attitudes not coinciding with teaching practices (e.g., Judson, 2006). In other words, being open and receptive to technology integration does not necessarily translate to meaningful technology usage in classrooms. Like other disciplines, culture and context are key variables to consider when examining teachers' use of technology in teaching and learning settings (EdTech Evidence Exchange, 2021). Ertmer and Ottenbriet-Leftwich (2010), for example, identified culture as a critical variable in teacher education, alongside knowledge, self-efficacy, and pedagogical beliefs. The authors argued that even if teachers have the appropriate knowledge and skills, feel confident in their abilities to integrate technology, and have strong pedagogical beliefs around technology integration, the context in which teachers work could still constrain or limit their individual efforts. In a similar vein, teachers who are reluctant to use technology and have little experience and knowledge might be confronted with outside pressures to introduce technology in their classrooms, which if met with positive results, could enact change in teachers' knowledge, attitudes, beliefs, and practices.

Non-STEM Domains

There is a wealth of knowledge around literacy and language instruction in early childhood settings. However, there is mixed evidence about the associations between language and literacy instruction and teachers' knowledge, beliefs, and instruction (Schachter et al., 2016). Whereas some studies revealed significant relationships among these constructs, other studies indicate inconclusive or inconsistent findings. For instance, Hammond (2015) found that although teachers believed knowledge about teaching reading was important to their role, their understanding of literacy skills (i.e., teachers' literacy knowledge) was generally low, and teachers who demonstrated high literacy knowledge did not always demonstrate these skills in their actual teaching. Professional development and coaching may improve teachers' knowledge and beliefs around literacy and language instruction to varying extents (Armstrong et al., 2008; Ottley et al., 2015).

In other non-STEM domains such as music and art, research supports the general argument that teacher knowledge, beliefs, and attitudes are interconnected. Kaleli (2020), for example, found a significant relationship between pre-service music teachers' self-efficacy beliefs and attitudes towards the teaching profession. In addition, Grauer (1998) found that at the beginning of a year-long art education program, general elementary teachers demonstrated less art content knowledge than specialized art teachers, which influenced the kinds of beliefs the two groups of teachers had about art and about teaching art. As teachers acquired disciplinary and pedagogical knowledge over the year, their beliefs also changed, indicating a complex, evolving relationship between knowledge and beliefs.

Summary

Understanding how teacher knowledge, attitudes, beliefs, and instructional practices are examined in other disciplines is useful for exploring these constructs in the context of early coding and robotics education. Findings regarding the relationships among the constructs are complex, often inconsistent, and dependent on the specific subject matter and measures used. However, there is consensus that each component of Ernest's (1989) model—teacher background characteristics (i.e., context), content knowledge, attitudes, instructional beliefs, and instructional practices—are important for understanding classroom teaching. Just as Ernest applied his model to the discipline of mathematics, this next section will introduce the discipline of focus for this dissertation: computer science.

Computer Science in K-12 Education

The K–12 Computer Science Framework (2016), co-developed with states and districts by the Association for Computing Machinery, Code.org, the Computer Science Teachers Association, the Cyber Innovation Center, and the National Math and Science Initiative, defines CS as "the study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications, and their impact on society" (p. 6). Coinciding with this definition, CS in this dissertation is viewed as a broad discipline, encompassing skill-based activities such as coding (used interchangeably with computing and programming), tangible technologies such as robotics, and computational ways of thinking and problem-solving (Tucker et al., 2003).

There are several key learning and developmental theories commonly cited in the CS education literature: constructionism, constructivism, and social constructivism. Seymour Papert, developer of the LOGO programming language in the late 1960s, was among the first to view coding as a creative learning activity that could and should be introduced to young children (Papert, 1980). His constructionist approach emphasized how individuals can acquire new knowledge by actively creating and sharing artifacts such as computer programs. Papert's ideas were heavily influenced by Jean Piaget's concept of constructivism. Piaget argued that children make sense of their world through the dynamic process of assimilation and accommodation. New information is acquired through experiences, which either gets incorporated into an existing schema or transforms the schema into a more nuanced model. Whereas Papert and Piaget's focus was on the individual, Lev Vygotsky's theory of social constructivism emphasized how learning is often fueled by interactions with adults or more capable peers. Other educational psychologists have since extended Vygotsky's theory to understand the role of scaffolding, or the ways in which the instructor or learning environment supports individuals in acquiring new knowledge (Ackermann, 2001). Social constructivism is key in the coding context because teachers' and students' varying experiences with CS influence how they can learn from one another in the classroom. Taken together, all three theories—constructionism, constructivism, and social constructivism—are foundational to understanding how and why CS has a rightful place in K-12 education.

In recent years, there has been an increasing national and international push for "computer science for all", which stems from the growing evidence base around K-12 CS education and external economic, social, and cultural factors (Fluck et al., 2016). Much attention has focused on the underrepresentation of girls and other minoritized students in the computing field, resulting in increased momentum around equity pedagogies in CS teaching and learning (Madkins et al., 2020; Sullivan, 2019). In the last 15 years, due to computer scientist Jeannette Wing's (2006) influential article in *Communications of the ACM*, an even larger focus has been placed on computational thinking. Wing resurfaced arguments from Papert (1980) and others that computational thinking was rooted in CS but involves a broad set of analytic and problem-solving skills, dispositions, and habits that are universally applicable and necessary for all children to learn. As such, current CS research and educational efforts have emphasized the need to introduce technologies and pedagogical approaches that engage students, as early as pre-kindergarten, to coding and computational thinking skills (Barr & Stephenson, 2011; Grover & Pea, 2013). This next section will present a summary of research on these early childhood efforts.

Coding, Robotics, and Young Children

Since LOGO, many programming languages and tools have been developed to introduce young children to foundational CS and computational thinking concepts. Yu and Roque (2018) conducted a survey of computational kits for young children and identified 34 physical, virtual and hybrid interfaces targeted for children ages seven and younger. Physical interfaces, like the KIBO robotics kit used in this dissertation, contain tangible objects that children can manipulate. Virtual or graphical interfaces are PC or mobile device-based applications without physical components, such as the Scratch and ScratchJr programming applications. Hybrid interfaces have both physical and virtual parts, such as Dash and Dot robots that use graphical programming blocks or Strawbies that uses tangible command tiles. Different coding interfaces offer different user experiences based on the design features of the tool, how the tool enables children to explore computational concepts and practices, and the range of activities and projects children can engage in using the tool (Strawhacker, Sullivan & Bers, 2013; Yu & Roque, 2018).

Across academia and industry, studies of young children's experiences with coding technologies reveal a variety of benefits for children's learning and development. Learning CS at a young age can support problem-solving and computational thinking skills (Barr & Stephenson, 2011; Clements & Sarama, 1997; Kafai & Burke, 2014; Seiter & Foreman, 2013; Wing, 2006). Children as young as four and five have shown increased understanding of algorithms, control structures such as repeat loops and conditionals, and debugging strategies after participating in introductory coding and robotics activities (Elkin, Sullivan & Bers, 2016; Strawhacker & Bers, 2019; Wohl, Porter & Clinch, 2015). Studies have also indicated that coding interventions can support children's positive behaviors and human virtues, such as curiosity, generosity, and perseverance (Bers, 2019, 2022).

Robotics and Tangible Interfaces

Of the plethora of coding technologies on the market, robotics has emerged as a successful tangible tool for young children. The term "robot" refers to an autonomous machine that can sense its environment, carry out preprogrammed tasks, and perform actions in the real world (Bers, 2008). Robotics kits consist of multiple physical parts (e.g., motors, sensors, structural and mechanical pieces) that can be assembled to create a functional robot. Physical parts and manipulatives extend developmentally appropriate practices already prioritized in early childhood education, such as the use of blocks and balls to promote children's fine motor skills and learning of mathematical concepts such as size and shape (Brosterman, 1997; Meacham & Atwood-Blaine, 2018). Several studies indicate that young children find robotics kits more inviting and engaging in comparison to unplugged (without a computing device) CS activities or screen-based coding platforms (Pugnali, Sullivan & Bers, 2017; Strawhacker & Bers, 2015; Wohl, Porter & Clinch, 2015). In the early childhood classroom, robotics also offers the possibility of engaging children in collaboration and peer discussion (Khanlari, 2016; Sullivan & Bers, 2017; Toh et al., 2016). For instance, one kit is often shared by multiple children, which encourages children to interact, share parts, and engage in dialogue about their joint activity. This dissertation study utilized the KIBO robotics kit, which is described next.

KIBO Robotics

The KIBO robotics kit (see Figure 2) is a tangible interface with modules, sensors, and color-coded programming blocks containing barcodes, text, and symbols. Children program the KIBO robot by using the barcode scanner embedded in the robot body to scan a series of barcode stickers on tangible wooden blocks. The kit contains light, sound and distance sensors and modules for exploring advanced programming concepts, as well as art platforms for decorating the robotic creations. KIBO has been used in research and educational settings around the world with children of diverse age groups and abilities (Albo-Canal et al., 2018; Bers, Gonzalez-Gonzalez, & Armas-Torres, 2019; Elkin, Sullivan, & Bers, 2016; Sullivan, Bers, & Mihm, 2017; Sullivan, Elkin, & Bers, 2015). Findings from prior studies indicate that children can use KIBO to learn computational concepts and engage in creative problem-solving skills. The KIBO-21 kit used in this study is comprised of the following 21 programming blocks: Begin, End, Wait for Clap, six blue Motion blocks, three yellow Light blocks, five orange Sound blocks, two Repeat blocks, and two If blocks.

Figure 2

KIBO-21 Robotics Kit

Summary

Technological advancements in the last several decades have led to a variety of coding tools that engage children not only as technological consumers, but also as producers of their own technological artifacts. Robotics kits for children, such as the KIBO robotics kit used in this study, have been highlighted as playful, engaging tools that promote children's learning of coding, as well as other developmentally appropriate learning goals. As robotics kits slowly made their way into classroom settings, many questions started to emerge about teacher knowledge and preparation, attitudes and beliefs around coding and robotics, and effective pedagogical practices. The following section explores these questions further.

Teaching Robotics in Early Elementary Classrooms

Over the last several decades, various studies have explored the impact of robotics on teachers and students in early elementary classrooms (Benitti, 2012; Cetin & Demircan, 2020;

Jung & Won, 2018; Toh et al., 2016). For the purposes of this literature review, this section highlights studies that specifically address the constructs of teacher knowledge, attitudes, beliefs, and practices.

Teacher Knowledge

Good teaching requires not only a substantial understanding of the subject matter, but also an adequate background in pedagogy (i.e., knowledge of how students learn) and in this context, technological expertise with robotics (Mishra & Koehler, 2006; Shulman, 1986). Most interventions for early elementary teachers, however, are short-term and focus mostly on developing teachers' CS content and robotics knowledge, rather than on pedagogical knowledge (Mason & Rich, 2019; Yadav et al., 2016). However, studies show that teachers need more training on how to incorporate coding technologies in their instruction. For instance, even if teachers understand what algorithms are, they report needing support on how to *teach* algorithms to their students (Giannakos et al., 2015). One important way of supporting elementary teachers' knowledge around coding and robotics, and how to teach coding and robotics to students, is professional development.

Professional development is defined as "any activity that is intended partly or primarily to prepare paid staff members for improved performance in present or future roles in the school districts" (Little, 1987, p. 491). Examples of such activities include organized workshops and university courses, as well as informal conversations with colleagues and teachers' self-reflections on their teaching practice. These examples illustrate the dynamic nature of professional development as ongoing, continuous, and embedded in teachers' daily lives (Desimone, 2009). The form of professional development often depends on the specific audience, for example, pre-service teachers who have yet to formally enter the teaching profession or in-service teachers currently in the education field. Various studies have examined the impact of professional development on pre-service and in-service elementary teachers (Bers et al., 2002; Chalmers, 2018; Jaipal-Jamani & Angeli, 2017; Kim et al., 2015; Mason & Rich, 2019; Menekse, 2015; Rich et al., 2017; Taylor, 2021). These studies have shown that hands-on workshops with opportunities for implementing and reflecting on robotics lessons can enhance teachers' overall robotics content and pedagogical knowledge. However, authors of these studies note that knowledge is only one component of professional development; teachers' attitudes and beliefs around coding and robotics should also be considered.

Teacher Attitudes and Beliefs

Papadakis and colleagues (2021) conducted a latent class analysis on 201 early childhood teachers' responses to a questionnaire about their attitudes towards educational robotics. The authors identified two clusters of teachers based on their survey responses. The first cluster was a homogenous group of teachers who shared positive attitudes towards educational robotics. The second cluster was a heterogenous group of teachers indicating inconsistent responses and expressing negative or skeptical views of educational robotics. The authors further discovered that teaching experience and age were negatively associated with membership in the first cluster, whereas educational robotics knowledge was positively associated with membership in the first cluster. These findings indicate that teachers have varied attitudes towards teaching robotics, and that their attitudes may be associated with specific teacher characteristics.

Connecting back to Ernest's (1989) theoretical model, teachers' attitudes are related to their beliefs (Valenzuela, 2019). Teachers who are receptive to CS education believe that engaging in coding and robotics learning can have positive effects on students' lifelong learning skills (Burke et al., 2020; Khanlari, 2016). Teachers who are more resistant to CS education have reported various internal and external barriers that impact their attitudes and beliefs. For instance, Khanlari (2016) and Kong (2019) highlighted lack of confidence, pedagogical content knowledge, support materials, and technology support as some of the most challenging factors for elementary teachers aiming to integrate robotics in their classrooms. Other studies have focused on teachers' self-efficacy, or belief in their capability to teach coding and robotics. For example, Jaipal-Jamani and Angeli (2017) found that a 12-week science methods course significantly enhanced pre-service teachers' interest and self-efficacy around teaching robotics. Furthermore, Ensign (2017) showed that participation in coding activities with students can promote teachers' confidence, suggesting a possible relationship between teacher beliefs and behaviors.

Teacher Behaviors

Observing how teachers integrate coding and robotics instruction in their classrooms have revealed effective pedagogical practices. Strawhacker et al. (2017) found that students performed significantly higher on programming assessments when teachers were responsive to student needs, flexible with lesson planning, competent with basic technological content, and mindful about promoting students' independent thinking. Students with diverse learning needs benefit from teacher modeling, scaffolded instruction, and collaborative problem-solving strategies (Israel et al., 2015). Teachers can scaffold learning using a variety of strategies: adapting tasks to make them more manageable, keeping students on task, and providing timely support for students before they get too frustrated and give up (Lye & Koh, 2014). Teachers also benefit from post-lesson reflections, adjusting their teaching strategies as necessary to meet the needs of students, and providing opportunities for students to document and share their work through design journals and discussions (Bers et al., 2002; Lye & Koh, 2014).

Summary and Gaps

Understanding how teachers can effectively integrate educational robotics in their classrooms requires a deep examination of their knowledge, attitudes, beliefs, and practices. Prior studies have uncovered some relationships among these constructs; however, there are some gaps in the literature that this dissertation aims to fill. First, there is a need for more qualitative studies that examine teachers' experiences from their own perspectives. For a teacher who has never experienced coding before, let alone be tasked with teaching coding to students, what forms of knowledge do they need and what might they lack? What are their attitudes regarding teaching coding and how do those attitudes shape their beliefs about coding education? These questions are best answered through qualitative methods such as interviews, which can enable teachers to reflect on different aspects of their experiences and attitudes that they may not necessarily communicate in a survey. Another important gap is the need to increase opportunities for teachers to actively engage in teaching coding. Although professional development is essential and can be an appropriate interventional tool to shape teacher knowledge, attitudes, and beliefs, more research is needed on teachers' instructional practices and how teachers implement what they have learned in their classrooms (Bers et al., 2013; Yadav et al., 2018). In this dissertation study, teachers were interviewed even before they received any formal training and participated in follow-up interviews until after they implemented a full-length robotics curriculum with students, providing a wide range to explore possible shifts in teachers' experiences and attitudes. The next chapter introduces the interventional tool and program presented in this dissertation.

Chapter Three: Statement of Problem

Although coding and robotics are an increasingly popular topic of interest in education research, teacher experiences and attitudes are often overlooked in the literature. This dissertation aimed to address this gap by drawing upon interview data from a sample of second grade educators in a U.S. public school district. To introduce the research questions for this dissertation, it is necessary to provide some context on the broader study from which data were collected and analyzed for this dissertation.

Coding as Another Language (CAL) Intervention

Many educators, researchers, and policymakers often associate coding as a problem-solving activity and thus position coding under the STEM umbrella, and rightly so, considering the high number of computing jobs in STEM-related fields. However, the teaching and learning of coding in early childhood is not about producing the next generation of computer scientists. Rather, the goal is to introduce children to new ways of thinking and expressing ideas through the creative and meaningful use of technology. With this goal in mind, coding is commonly positioned as a "literacy of the 21st century". The term *literacy* is often invoked to emphasize the importance of coding in our modern technology-rich world. After all, at the turn of the twentieth century, it was difficult to imagine achieving economic independence or participating in civic society without knowing how to read and write. Similarly, it may soon be difficult to succeed in the modern world without some knowledge of coding, or at the very least, a foundational understanding of the computational processes involved in computing.

This metaphor of coding as a literacy holds meaning for several other reasons. First, there are multiple points of alignment between the two disciplines. Natural languages (like English or Spanish) and artificial languages (like Scratch or the KIBO block language) are similar in that both are symbolic systems of representation with communicative and expressive functions. Children can use natural and artificial languages to create personally meaningful artifacts that can be shared with others. Although there are important differences between the two (e.g., a person might be able to decipher the meaning of a poorly written message with many grammatical errors, but a computer cannot interpret imprecise instructions), the points of alignment are often overlooked. Second, early elementary teachers are responsible for teaching a wide range of subjects, which leaves little time to allocate towards coding unless it is integrated with other content areas. Particularly in the U.S. education system, English/Language Arts and Mathematics are the two most emphasized subjects in the early grades. If reading and writing using natural languages can be taught in tandem with reading and writing using artificial languages, the pedagogical overlaps might serve useful to both disciplines.

The pedagogical approach and curriculum designed for this study, “Coding as Another Language” (CAL), stemmed from this premise that the disciplines of CS and literacy/language have key areas of overlap (Bers, 2019; Govind et al., 2021). The CAL curriculum is comprised of lesson plans that highlight the similarities and differences between natural languages and artificial programming languages with respect to the creation process, their syntax and grammar, and their potential to empower individuals. For example, students are presented with the design process and writing process side-by side (both of which are creative processes that involve iterative planning, testing and evaluation) and then engage in a discussion around the similarities and differences between programmers and writers. Table 1 displays seven key CS ideas, adopted from Bers’ (2018) seven powerful ideas of computational thinking in early childhood, and how these correspond to complementary concepts in English/Language Arts. In this dissertation study, the CAL connections are operationalized using the KIBO robotics kit, hence the name of the curriculum "Coding as Another Language-KIBO" or CAL-KIBO.

Table 1

Coding as Another Language (CAL) Connections

Powerful Ideas of Computer Science and Computational Thinking	Corresponding Concepts in Literacy and Language	Description
Algorithms	Sequencing	Emphasis on "order matters," and on how complex tasks can be broken down into step-by-step instructions in a logical way.
Design Process	Writing Process	Creative, iterative, cyclic processes that involve imagining, planning, revising, and sharing, often with different starting points.
Representation	Alphabet and Letter-Sound Correspondence	Sounds and symbols have different attributes (e.g., color, shape, sound) that can be used to represent something else.
Debugging	Editing and Audience Awareness	Systematic analysis, testing, and evaluation to improve communication to the intended audience (computer or person). Whenever miscommunication occurs, the programmer or writer uses a variety of strategies to solve the problem.
Control Structures	Literary Devices	Advanced strategies that determine how a set of ideas or commands are executed.
Modularity	Phonological Awareness	Decomposition, or breaking down a complex task into smaller modules and re-using those modules in different ways.
Hardware/Software	Tools of Communication and Language	Communicating abstract ideas through tangible means. Just like hardware and software work together, the expression of thoughts through language requires a medium for communicating to the outside world, such as spoken or written word.

Preliminary findings from piloting the CAL pedagogical approach and CAL-KIBO curriculum have been promising. For instance, with a subset of second graders from the larger

NPS study, we explored the relationship between 132 students' standardized literacy scores (Phonological Awareness Literacy Screening, or PALS) at the beginning of the school year and their KIBO programming skills at the end of the CAL-KIBO curriculum. The results indicated evidence for a weak, positive correlation ($r = 0.3$) between students' literacy and programming scores, suggesting that there may be an overlap between language and literacy ability in children on the one hand and the ability to learn computational concepts on the other (Hassenfeld et al., 2020). Bers et al. (2021) expanded on these preliminary findings with the full dataset of first and second grade students in the NPS district. We explored the relationship between students' standardized literacy scores (PALS and the Developmental Reading Assessment, or DRA) and students' computational thinking skills (as measured by the unplugged *TechCheck* assessment). Findings indicated that students' literacy scores at the beginning of the year significantly predicted their post-curriculum *TechCheck* scores, when controlling for students' pre-curriculum baseline *TechCheck* scores. In addition, Relkin, de Ruiter & Bers (2020) examined changes in computational thinking skills in first and second grade students who had been exposed to the CAL-KIBO curriculum compared to children who did not. Over the course of the study, children who received CAL-KIBO improved on their computational thinking skills (as measured by *TechCheck*), whereas the control group did not. These preliminary findings indicated that the CAL-KIBO curriculum was effective in teaching young children not only to code, but also to think computationally.

Preliminary findings on NPS teachers also showed interesting trends. Relkin & Bers (2020) found that teachers' post-training KIBO knowledge performance (as measured by the TACTIC-28 assessment) was a significant predictor of students' post-curriculum computational thinking performance, suggesting a possible relationship between teacher knowledge and instructional behaviors (which could have contributed to improved student outcomes). When we examined a subset of second grade teachers' survey responses over the course of the study (pre-training to post-curriculum), our analyses showed that teachers reported increases in their coding knowledge and attitudes after the training and even more so after implementing the CAL-KIBO curriculum. However, the variability in teachers' responses to the open-ended items at the end of the survey spurred questions about teachers' unique strengths and challenges they experienced over the course of the CAL-KIBO program (Govind & Bers, 2020). Some of these reported strengths and challenges were discussed in Bers et al. (2021), for instance, teachers' varying perceptions of integrating coding and literacy, and how teachers navigated issues related to time, space, and materials differently based on individual and contextual factors. This dissertation expands on these findings by focusing on a subset of second grade teachers who spoke with our research team in semi-structured interviews at various points of the study. Specifically, this dissertation set out to address the following questions:

1. What are second grade teachers' experiences and attitudes around coding and robotics education?
2. What impact, if any, does participation in the Coding as Another Language (CAL) – KIBO program have on teachers' experiences and attitudes?

As is common in qualitative research, initial research questions can be tentative and exploratory but are necessary for articulating the primary focus of a study (Agee, 2019; Creswell, 2007). Over the course of analysis, these research questions can evolve based on the nature of findings and how they relate to the initial questions that were asked. Subsequent chapters will detail how the two research questions initially set forth in this dissertation were modified slightly to reflect the nature of findings.

Chapter Four: Methodology

Larger Study Context

This dissertation draws upon data collected from a two-year research collaboration between the DevTech Research Group at Tufts University, led by principal investigator Prof. Marina Umaschi Bers, and the Norfolk Public School (NPS) District (Bers et al., 2021). The NPS district is a mid-sized city school district located in southeastern Virginia close to Chesapeake Bay, home of the world's largest naval station. According to the National Center for Education Statistics, during the 2018-2019 school year, 72.5% of the district's students were eligible for free or reduced-price lunch, 2.8% were English language learners, and 14.4% of students had an Individual Education Plan (IEP). The DevTech-NPS research collaboration was funded by a U.S. Department of Defense Education Activity (DoDEA) grant entitled "Operation: Break the Code for College and Career Readiness". The federal grant supported computer science education and socioemotional learning opportunities in a subset of the district's elementary schools that serve a high number of military-connected students.

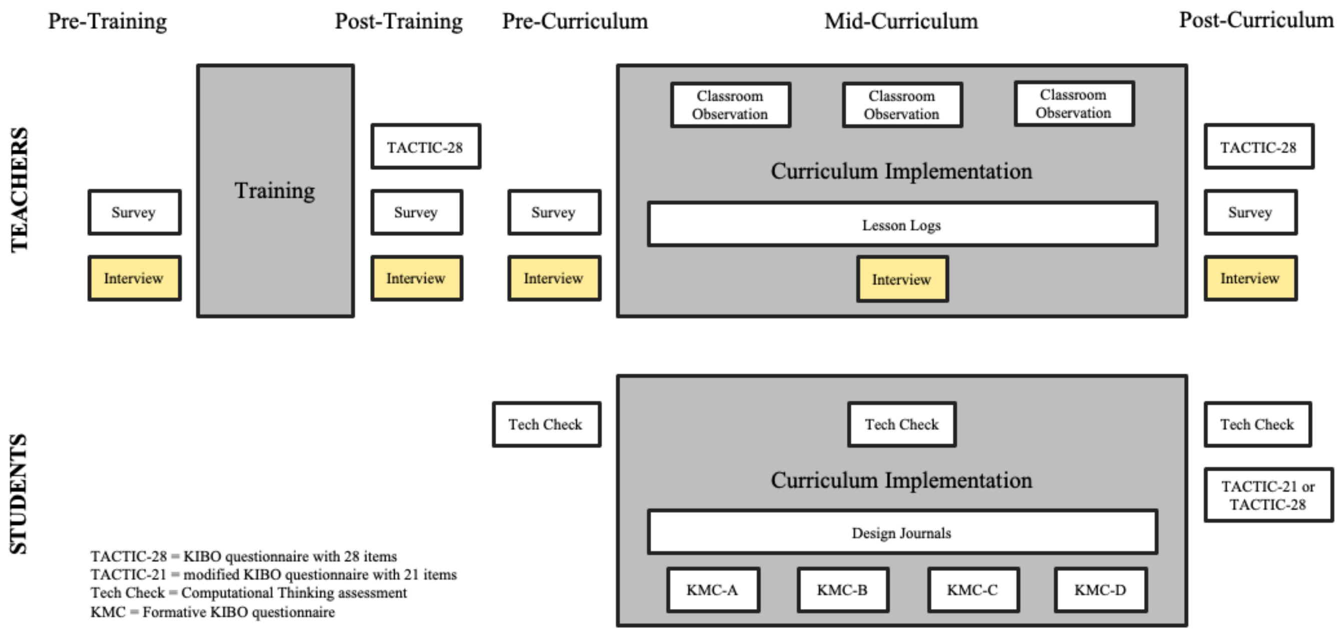
The goal of the research collaboration was multi-fold: 1) to identify best practices for professional development and curricular resources for teaching coding through KIBO robotics in early elementary classrooms; 2) to test and validate developmentally appropriate assessments that capture young children's acquisition of coding and computational thinking skills; and 3) to explore how teaching coding with a literacy-oriented approach (i.e., Coding as Another Language) can contribute to children's growth in coding, computational thinking, and literacy skills. Findings from this dissertation will contribute to the first goal of this larger study. Findings contributing to the second and third goals have been previously disseminated in peer-reviewed journals (Bers et al., 2021; Hassenfeld et al., 2020; Relkin et al., 2020).

The two-year research collaboration first began with second grade classrooms during the 2018-2019 school year and subsequently with first grade classrooms in October 2019. Although plans were initiated to continue the project with kindergarten classrooms in March 2020, no data were collected beyond the professional development training due to the onset of the COVID-19 pandemic. Between each phase of data collection, the research team adapted curricular resources and study measures based on feedback from participants and the research team. The second grade sample was split into two waves of data collection: one set of four schools participating in November 2018 – March 2019 (Wave 1), and the second set of four schools participating in March 2019 – May 2019 (Wave 2). Schools were randomly assigned to Wave 1 or Wave 2 by the research team in collaboration with the NPS district coordinator. The first grade sample participated altogether and began in October 2019. Due to competing priorities in first grade, one of the eight schools decided not to participate, and another school terminated their participation about halfway through the study. Thus, these two schools were removed from the study. This dissertation focuses on the second grade sample from the remaining six elementary schools.

Multiple types of data were collected from teachers and students at various points of the training and curricular intervention (see Figure 3). This dissertation analyzes a subset of data collected from this larger study, focusing specifically on the teacher interviews (highlighted in yellow in Figure 3). The research team also conducted teacher surveys, classroom observations, assessments of children's coding and computational thinking skills, and other measures as part of this larger study. These additional measures are not examined for this dissertation but are explained further in Appendix C.

Figure 3

Study Procedure



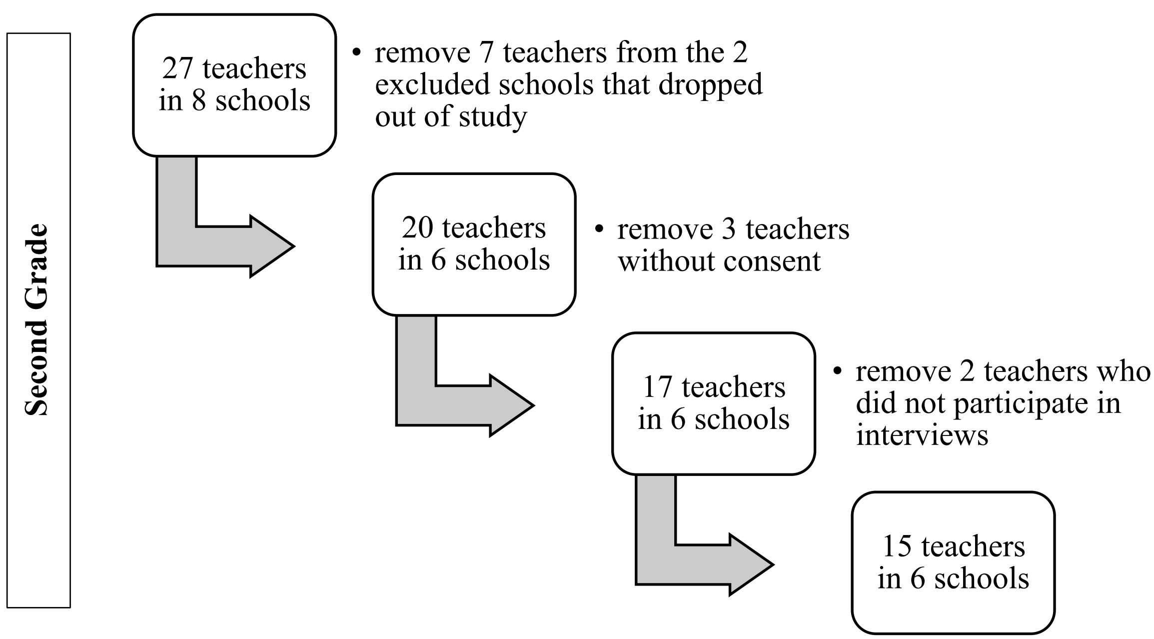
All data collection procedures were approved by the Tufts University Institutional Review Board (protocol #1810044). Initial and ongoing consent was obtained from participating teachers. All teachers, regardless of consent, were given the opportunity to attend the full-day training and implement the CAL-KIBO curriculum in their respective classrooms. Only consented teachers participated in the research activities (e.g., interviews, surveys, etc.). An approved alternate informed consent process was used for student data collection. Families were sent home a detailed letter that explained all research procedures involving their child. Families choosing to opt their child out of research participation were asked to sign and return the letter to their classroom teacher (later sent to the research team), in which case the teacher provided the child with alternate activities during the periods of data collection. Verbal assent was obtained from children prior to all research activities. No opt-out forms were signed by families and delivered to the research team, resulting in a high child participation rate.

The Analytic Sample

The flow chart in Figure 4 depicts how the analytic sample of teachers was obtained. A total of 27 second grade teachers from eight schools were initially invited to attend the full-day professional development training. As mentioned above, two schools were excluded from analysis due to dropping out of the study before it was fully completed. In addition, teachers who did not consent to research participation nor participate in any of the interviews were also removed from the sample. Thus, the analytic sample for this dissertation consists of *N* = 15 second grade teachers from six elementary schools in the NPS district.

Figure 4

Flow Chart of Analytic Sample



Comparative tests were conducted to check for possible differences between the analytic sample of 15 teachers and the 12 teachers excluded from the analytic sample. The following demographic variables and teacher measures were used to compare the two groups: 1) teacher's race/ethnicity, 2) years of teaching experience, 3) number of lesson logs completed by the teacher, and 4) teacher's post-training TACTIC-28 score (see Appendix C for further descriptions of these measures). The alpha value for determining statistical significance was set to .05. Results of these comparative tests indicated the following:

1) A Chi-Square test of independence indicated no significant difference in race/ethnicity between the two groups, $X^2(4, 27) = 3.60$, $p = .463$.
2) A Chi-Square test of independence indicated no significant difference in years of teaching experience between the two groups, $X^2(4, 27) = 4.37$, $p = .498$.
3) An independent-samples *t*-test indicated no significant difference in the mean number of lesson logs completed by teachers in each group, $t(24) = 0.71$, $p = .485$. On average, teachers in the analytic sample completed 7.2 lesson logs, and teachers in the non-analytic sample completed 5.9 lesson logs.
4) An independent-samples *t*-test indicated no significant difference in post-training TACTIC-28 scores for teachers in each group, $t(6.738) = 0.29$, $p = .783$. The mean scores for teachers in the analytic and non-analytic samples were 21.5 and 21.0 points, respectively.

Altogether, these findings suggest that the teachers excluded from analysis are not that significantly different from the teachers included in the study, which contributes to the overall generalizability of study findings. Comparative tests were also conducted to check for possible differences between the 272 students in the analytic sample and the 279 students excluded from the analytic sample. The following variables were used to compare the two groups: 1) student's fall PALS summed score, and 2) student's post-curriculum TACTIC-28 score (see Appendix C for further descriptions of these measures). The alpha value for determining statistical significance was again set to .05. Results of these comparative tests indicated the following:

1) An independent samples *t*-test indicated no significant difference in fall PALS summed scores between the two student groups, $t(496.703) = 1.50$, $p = .135$. The mean fall PALS summed scores for students in the analytic sample ($n = 271$) and non-analytic sample ($n = 247$) were 44.52 and 46.53 points, respectively.
2) An independent samples *t*-test indicated no significant difference in TACTIC-28 scores between the two student groups, $t(448) = 0.38$, $p = .703$. The mean TACTIC-28 scores for students in the analytic sample ($n = 234$) and non-analytic sample ($n = 216$) were 18.31 and 18.17 points, respectively.

Table 2 presents a summary of the analytic sample, which includes any second grade classroom teacher who consented to research participation and completed at least one interview. All 15 teachers identified as female. Ten teachers identified as White (66.7%), two as Black or African American (13.3%), one as Hispanic, Latino, or Spanish origin (6.7%), and one as Asian (2.8%). One teacher did not respond to this demographic question. Five teachers identified having 6-10 years of teaching experience (33.3%), four teachers with over 15 years (26.7%), two teachers with 11-15 years (13.3%), two teachers with 0-3 years (13.3%), and one teacher with 4-5 years (6.7%). One teacher did not respond to this question.

Table 2

Sample Demographics

School	**School Characteristics**		**Second Grade**	
	Range of Grade Levels	**Military-Connected Students (%)**	**Teachers (*n*)**	**Students (*n*)**
School C	K-5	39.8	1	16
School T	PK-5	67.1	2	39
School S	PK-5	76.5	4	61
School F	PK-2	12.1	2	38
School G	K-8	12.8	2	37
School B	PK-5	14.9	4	81
Total			**15**	**272**

Each classroom consisted of 14-22 students whose families did not opt them out of research participation and are also included in the student analytic sample. Student gender and race/ethnicity information were collected from administrative data files shared by the district. Of the 272 students in the student analytic sample, there were 122 girls (44.9%) and 150 boys (55.1%). The race/ethnicity distribution was as follows: 116 students were identified as Black or African American, 97 students as White, 30 students as Hispanic, four as Asian, one as Pacific Islander or Hawaiian, one as Native American Indian or Alaskan, and 23 as Two or More Races. Table 3 shows the race/ethnicity breakdown for the student analytic sample, entire NPS district, Virginia K-12 schools, and U.S. public K-12 schools in the 2018-2019 school year. From these data, it is reasonable to conclude that the race/ethnicity distribution of the analytic sample resembles that of the district but differs from the broader state and national race/ethnicity distributions, particularly with respect to the relative proportion of Black, Hispanic, and White students.

Table 3

Race/Ethnicity Comparisons Between Sample and Broader Contexts

Race/Ethnicity	Percentages (2018-2019)			
	Analytic Sample	Norfolk Public Schools	Virginia K-12 Schools	U.S. Public K-12 Schools
American Indian/Alaska Native	0.4	0.4	0.3	1
Asian or Asian/Pacific Islander	1.5	2.3	7.1	5
Black	42.6	58.9	22.2	15
Hispanic	11.0	10.2	16.2	27
Native Hawaiian/Pacific Islander	0.4	0.3	0.2	<1
Two or More Races	8.5	6.5	5.7	4
White	35.7	21.4	48.4	47

Professional Development

Prior to curriculum implementation, classroom teachers attended a full-day professional development training facilitated by 3-4 members of the DevTech research team. All trainings followed a similar agenda: introductions, discussions around teaching coding in early childhood, hands-on play with the KIBO robotics kit, overview of the CAL-KIBO curriculum, and a deep-dive into CAL-KIBO lesson activities. During the training, classroom teachers were seated next to their colleagues, which enabled teachers to learn alongside one another and discuss plans for curriculum implementation. The schools' designated coding facilitators and instructional technology resource teachers (ITRTs) participated in the same training as classroom teachers. Coding facilitators occupied full-time roles at the school (e.g., classroom teacher, librarian, media specialist, etc.) but were given additional responsibilities to coordinate the school's coding-related initiatives. ITRTs were district staff members responsible for supporting classroom teachers at one or more schools with technology-related teaching and learning activities.

After the training, classroom teachers engaged in ongoing professional learning, which included weekly emails and video tutorials for every CAL-KIBO lesson, on-demand coaching with the research team, and additional in-person support as requested from school staff. The weekly emails included resources and tips for upcoming lessons, salient quotes from teachers' lesson logs from the previous week, video tutorials and slide decks, and reminders regarding upcoming research activities. The video tutorials were 2-3-minute overviews summarizing the purpose of each lesson activity. On-demand coaching was provided in the form of email, phone call, and in-person conversations during site visits. School staff (e.g., district coordinator, ITRTs, coding facilitators) also met with teachers periodically to assist with lessons or offer additional guidance. Teachers were asked about their use of these various professional learning supports during their semi-structured interviews.

CAL-KIBO Curriculum

The CAL-KIBO curriculum implemented in second grade classrooms consisted of 12 one-hour lessons (see Table 4 for curriculum overview and Appendix D for the full curriculum). Each lesson was divided into multiple activities such as songs (e.g., the Engineering Design Process Song), movement activities (e.g., Programmer Says), writing activities (e.g., How-To prompt), and coding activities (e.g., Program the Hokey Pokey). Before introducing any KIBO blocks, the beginning lessons first established programming as an activity for communication and self-expression. The lessons then introduced the KIBO robot and its programming blocks, weaving between the new block functions and the connections back to composition, self-expression, and language. These connections were mediated through the focus book *Where the Wild Things Are* by Maurice Sendak. For example, Lesson 6: What Did Max Sense reviewed the way the protagonist, a little boy named Max, uses his five senses in the story. This activity led into the introduction of KIBO sensors and corresponding sensor blocks. Teachers were guided to lead a discussion comparing the "poetic" language used in the story to describe Max's senses and the contrasting command language needed for sensor blocks in the KIBO programming language. The final project in the curriculum invited students to create their own version of the "Wild Rumpus". Students first described their own Wild Rumpus in writing and then programmed the actions using their robots. Throughout the lessons, students recorded their ideas in individual design journals to document their learning process and put their emerging literacy skills into practice.

Because the CAL-KIBO curriculum was first implemented in second grade and then first grade classrooms, several changes were made to the lessons to incorporate second grade teachers' feedback and to adapt for younger children. One change was extending the final lesson into three lessons to allow students more time to plan and create their final projects. The compositional activity for the "Wild Rumpus" project was also removed because of first graders' limited writing skills, which provided students more time to program and share their robotic creations. Another change was omitting conditional statements from the first grade curriculum, so that students could explore different repeating patterns and loops. Finally, teacher and classroom support materials were expanded for the first grade curriculum. Whereas second grade teachers received printed curriculum binders, lesson tutorial videos, weekly emails, and lesson slides, first grade teachers additionally received vocabulary slides, KIBO slides, and access to all curriculum materials through a public website. The original second grade CAL-KIBO curriculum, prior to making adaptations for first grade, is included in Appendix D.

Table 4

Coding as Another Language (CAL)-KIBO Curriculum Overview

Lesson	Lesson Objectives
Lesson 1: Foundations	• Define engineer and understand that there are different types of engineers • Compare and contrast the Design Process and Writing Process • Use the Design and Writing Processes to write a set of instructions for making something
Lesson 2: Technological Tools – Robots	• Identify characteristics of a robot • Compare human languages and programming languages • Create a simple algorithm using the KIBO programming blocks
Lesson 3: Sequencing	• Understand why order matters when programming a robot or telling a story • Identify the different parts of the KIBO robot
Lesson 4: Programming	• Tell and retell a story clearly and effectively • Identify common errors with scanning KIBO programs and troubleshoot them • Practice scanning programs with KIBO • Learn strategies for debugging and editing
Lesson 5: Debugging	• Identify common errors with scanning KIBO programs and troubleshoot them • Practice scanning programs with KIBO • Learn strategies for debugging and editing
Lesson 6: Cause and Effect – Level 1	• Distinguish between human senses and robot sensors • Use the KIBO Sound Sensor with its appropriate Wait for Clap block • Record a sound clip successfully using the Sound Recorder module and Sound Recorder blocks
Lesson 7: Cause and Effect – Level 2	• Program KIBO to sing and dance to the "If You're Wild and You Know It" song
Lesson 8: Repeat Loops – Level 1	• Identify patterns in code sequences and rewrite codes using repeat loops • Use KIBO number parameters to make a program that loops a specific number of times • Understand how repetition is used in stories and songs
Lesson 9: Repeat Loops – Level 2	Second Grade: • Compare and contrast human senses and robot sensors

	• Successfully test a KIBO program using the Distance and Light sensors Objectives modified for First Grade: • Compare and contrast human senses and robot sensors • Successfully test a KIBO program using the Light sensor
Lesson 10: Repeat and If Statements	Second Grade: • Successfully test a conditional KIBO program using the Distance and Light sensors • Identify situations that would require an If statement or a Repeat loop Objectives modified for First Grade: • Successfully test a KIBO program using the Distance sensor
Lesson 11: Final Project – Writing the Wild Rumpus Composition	• Utilize the Writing Process by writing their Wild Rumpus composition • Decide which of their ideas can and cannot be translated into KIBO programs
Lesson 12: Final Project – Coding the Wild Rumpus (First Grade's final project split into three individual lessons)	• Demonstrate the Design Process in full by planning, designing, and creating a final KIBO project • Share final projects with peers, family, and community members • Identify and show appreciation to those who have helped them with their final projects

Semi-Structured Interviews

Classroom teachers were invited to participate in semi-structured interviews (Patton, 2015) at various times during the research study: pre-training, post-training, pre-curriculum, mid-curriculum, and post-curriculum. A semi-structured approach for the interviews enabled researchers to ask questions as a starting point for conversation and then follow up with relevant questions and prompts. The pre-training interview primarily served as an introduction for teachers to the research study and for researchers to gain initial insight into teacher and classroom demographics. The post-training interview included questions such as "What are you looking forward to most/least about KIBO and this curriculum? What kind of support would be most or least helpful?" The pre-curriculum interview (conducted several weeks or months after the post-training interview depending on the wave of data collection) similarly asked questions about teachers' reactions and kinds of support needed. The mid-curriculum interview was conducted during teachers' curriculum implementation and focused on teachers' current experiences and attitudes, kinds of support they found most and least helpful, and pedagogical strategies they found most useful. The post-curriculum interview was conducted after teachers completed all of the lessons, covering questions such as "What have been your successes/challenges of implementing this curriculum? How did these activities fit into the rest of your classroom curriculum? What support resources were most/least helpful?" Although some questions remained the same across timepoints, other questions were varied to focus on teachers' experiences and attitudes around that specific point in time (see Appendix A for the full interview protocol). Interviews were audio-recorded and later transcribed using an online transcription service. Any gaps in the transcripts were filled by research assistants who reviewed the audio files manually.

There are several limitations and considerations to note about the interview data collection process: semi-structured format, attrition over the course of the study, and variability of interview duration. With a semi-structured interview format, the interviewer may use her discretion to probe further on specific interview questions. Thus, each interview follows a slightly different line of questioning, which could lead to participants reflecting on different aspects of their experiences. In addition, due to the longitudinal nature of the study and elementary teachers' busy schedules, it was expected to have some amount of missing data and sample attrition. To minimize this limitation to the extent possible, efforts were taken by the research team to send interview appointment reminders and schedule interviews around teachers' planning time. Table 5 displays the total number of interviews collected across the five timepoints, as well as the mean interview duration for each timepoint. The average total interview time per teacher over the course of the study was 31.1 minutes. The variability in interview duration can be attributed to the kinds of questions asked at each timepoint, teachers' willingness to share their thoughts and experiences, and interviewers' ability to probe and ask follow-up questions to obtain more in-depth information.

Table 5

Summary of Interview Data Collection

Timepoint	Count	Duration	
	n	***M* (min)**	***SD* (min)**
Pre-Training	4	12.4	2.7
Post-Training	8	1.6	0.7
Pre-Curriculum	10	12.7	4.7
Mid-Curriculum	6	17.2	4.7
Post-Curriculum	10	17.4	5.9
Total	**38**	**31.1**	**19.4**

Analytic Procedure

Whereas the larger study utilized a combination of quantitative and qualitative measures, the focus of this dissertation is qualitative. I used Braun and Clarke's (2006) six-phase thematic analysis approach to analyze the data collected from the semi-structured educator interviews. Thematic analysis is defined as "a method for identifying, analyzing, and reporting patterns (themes) within data" (p. 79). This approach involves reading through the data multiple times, generating initial codes, combining codes into overarching themes, reviewing the themes to check if they align with codes and the overall research questions, clarifying themes in greater detail, and writing up findings while referring to the data to ensure that the findings provide an accurate representation of the collected data.

There are multiple advantages to using thematic analysis. For instance, the approach offers flexibility in how data are coded and can provide a useful summary of key features in a large body of data, such as the one analyzed in this dissertation. The approach also enables the researcher to take ownership of how the codes and themes are interpreted, allowing for social and psychological interpretations of the data. In doing so, the researcher takes an active role in identifying themes and patterns, as opposed to the common misconception that themes "passively emerge" from the data (Braun & Clarke, 2006, p. 80). Proponents of thematic analysis argue that themes can be appropriately identified and interpreted when there is an existing theoretical framework that anchors the analytic claims that are made. In addition, they emphasize the need for a clear and reliable analytic procedure. The following section details the analytic procedure for each of the two research questions, how codes were checked for consistency using multiple coders, and the software used for each step of the analysis. The chapter concludes with slightly modified research questions that better represent the essence of study findings.

Research Question One

The first research question that this dissertation aimed to address was "What are second grade teachers' experiences and attitudes around coding and robotics education?" To answer this question, I read through each interview transcript and highlighted statements that reflected educators' experiences and attitudes. These highlighted statements were then transferred to a spreadsheet for initial coding. Initial codes were compiled into categories and revised over time as new codes were generated. Codes were also informed by reviewing the literature and ensuring there was alignment between generated codes and existing theoretical frameworks. Once the codebook was finalized (see Appendix B for the full codebook), the dataset was transferred to a qualitative coding software for further analyses and data visualizations. Final codes were aggregated by teacher, school, and timepoint (Pre-Training, Post-Training, Pre-Curriculum, Mid-Curriculum, and Post-Curriculum).

Research Question Two

The second research question that this dissertation aimed to address was "What impact, if any, does participation in the Coding as Another Language (CAL) – KIBO program have on teachers' experiences and attitudes?" To answer this question, I examined the distribution of codes for each teacher and by school and timepoint. Interesting patterns and trends were explored further to derive themes. Teacher background characteristics such as school and years of teaching experience were also used to explore specific patterns. There was little variation in teachers' post-training TACTIC-28 scores, so this characteristic was not used to explore patterns. Code frequencies were examined to understand the relative distribution of codes across time, which may demonstrate changes in educators' experiences and attitudes at the level of the group. Due to missing data and sample attrition as previously described, longitudinal analyses at the individual level were not performed. However, case studies are used to describe common patterns and themes.

Reliability of Codes

To ensure reliability of generated codes, the full interview dataset was also independently coded by another graduate research assistant. This assistant and I met on a regular basis to review the codes, clarify inconsistencies, and discuss how the codes could be combined into broader themes. Inter-rater reliability was measured using the following formula described by Miles and Huberman (1994): reliability = number of agreements / (number of agreements +

disagreements). To ensure sufficient agreement between coders and mitigate the risk of interpretative bias, we set a goal of at least 85% agreement of 95% of the codes (McAlister et al., 2017). Percentage agreement was calculated to be 87% between the two coders.

Analysis Software

Semi-structured interviews were hosted and recorded using FreeConferenceCall.com. Teachers joined a private phone line with a passcode and were asked for permission before the interviewee started recording the call. Interview recordings were transcribed using manual transcription services from Scribie.com. Highlighted statements from the interview transcripts were populated into Microsoft Excel for initial coding and later imported into NVivo 12 for further analyses. Comparative tests conducted to understand the analytic sample of teachers and students were performed using IBM SPSS Statistics Version 26.

Evolution of Research Questions

It is not uncommon in qualitative studies to modify research questions as they may evolve over the course of analysis (Agee, 2009). In this study, after examining the interview transcripts and understanding how different teachers shared different aspects of their experiences and attitudes, it was evident that the initial research questions did not fully align to the identified themes. Thus, the research questions were modified slightly. The first research question was revised to reflect teachers' experiences and attitudes as part of the specific context of participating in the CAL-KIBO program. The focus of the second research question shifted to unpacking the factors that might explain the variance in teachers' experiences and attitudes.

The two revised research questions were as follows:

1. What were second grade teachers' reported experiences and attitudes over the course of their participation in the Coding as Another Language (CAL) – KIBO program?

2. To what extent did teachers' experiences and attitudes vary, and what kinds of individual and contextual factors might explain these differences?

Chapter Five: Results and Discussion

Research Question One: What were second grade teachers' reported experiences and attitudes over the course of their participation in the Coding as Another Language (CAL) – KIBO program?

Following Braun and Clarke's (2006) process for conducting thematic analysis, the initial steps involved reading through the data multiple times and generating codes and sub-codes, which were then grouped into categories. Using Ernest's (1989) theoretical model as a primary framework, the following overarching categories were identified: knowledge, attitudes, beliefs, and behaviors. An additional category was created to capture the type and quality of support that teachers reported either receiving or wanting over the course of their participation in the CAL-KIBO program.

Figure 5 displays the relative distribution of codes across all five categories (knowledge, attitudes, beliefs, behaviors, and support) for the full dataset of teachers across the various timepoints. Code frequencies and percentages across all teachers for the five timepoints are reported in Table 6. Support was one of the most discussed topics in the interviews, specifically with teachers sharing feedback on the curriculum and reflecting on the supports provided by the school and research team. Holistically, teachers shared mostly positive attitudes and beliefs about coding and robotics and expressed predominantly positive attitudes about integrating coding and literacy. Teachers reported several challenges, specifically around lesson implementation, KIBO organization, and coding and robotics knowledge. In addition, teachers described various pedagogical strategies used to facilitate students' learning, collaborative learning strategies being the most frequently mentioned. The following sections describe these five categories in more detail. The full codebook with example teacher quotes is provided in Appendix B.

Figure 5

Visual Map of Teachers' Reported Experiences and Attitudes

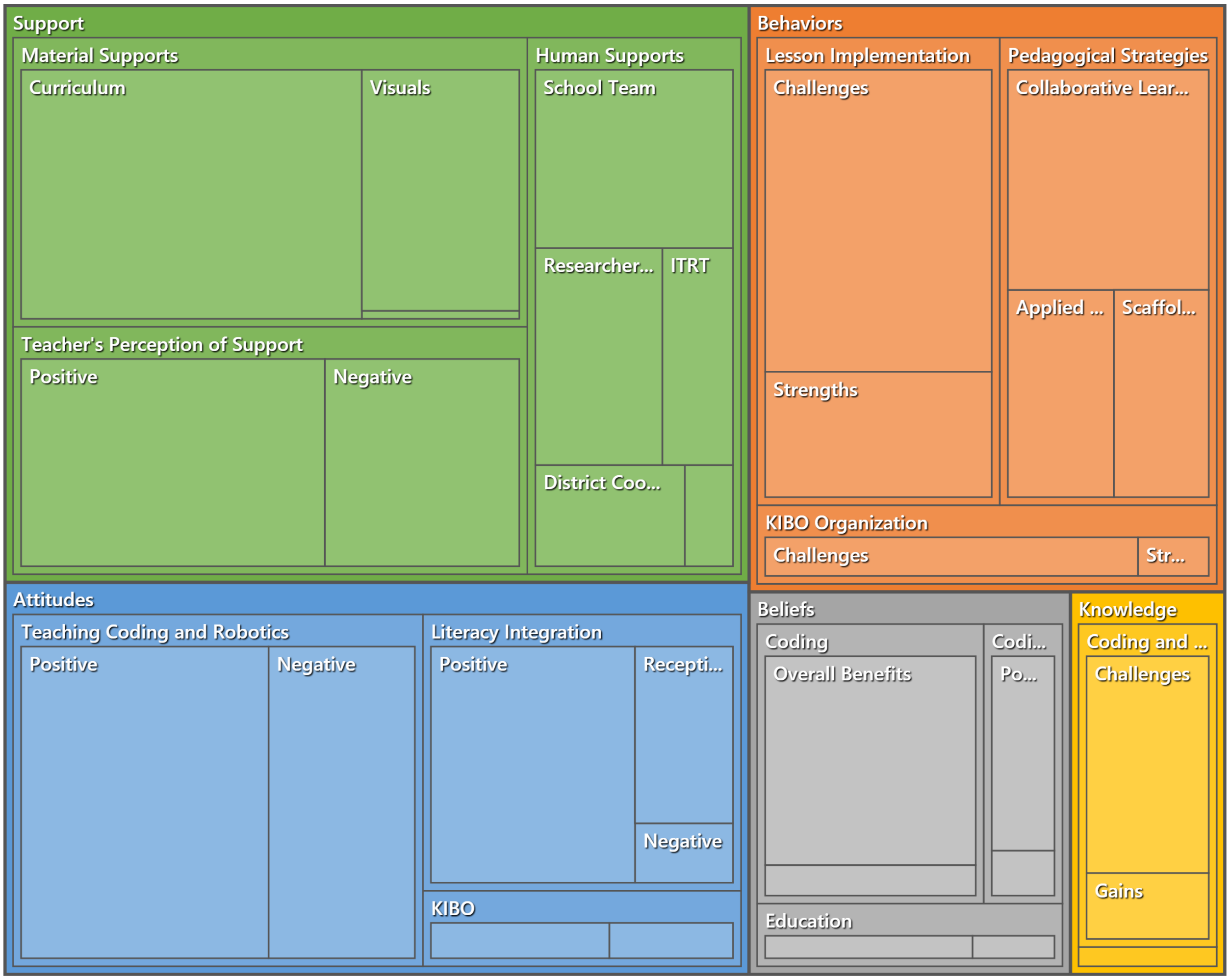


Table 6

Code Frequencies and Percentages Across All Teachers and Timepoints

Code Categories	Sub-Categories	*n*	%
Knowledge		**36**	**100**
Coding and Robotics	Gains	8	22.2
	Challenges	26	72.2
Teaching	Classroom Management	2	5.5
Attitudes		**181**	**100**
KIBO	Positive	10	5.5

	Negative	7	3.9
Teaching Coding and Robotics	Positive	64	35.4
	Negative	38	21.0
Literacy Integration	Positive	42	23.2
	Negative	5	2.8
Non-Literacy Integration	Positive	15	8.3
Beliefs		**76**	**100**
Coding	Overall Benefits	40	52.6
	Benefits for Young Children	6	7.9
Coding and Inclusion	Positive	13	17.1
	Negative	3	3.9
Education	Learning as Ongoing, Iterative Process	10	13.2
	Fun	4	5.3
Support		**179**	**100**
Human Supports	(all)	78	43.6
Material Supports	Visuals	31	17.3
	Curriculum (format and structure)	27	15.1
	Curriculum (lesson activities)	42	23.5
	Arts and Crafts Supplies	1	0.6
Teachers' Perception of Support	Positive	53*	-
	Negative	34*	-
Behaviors		**173**	**100**
Pedagogical Strategies	Collaborative Learning	36	20.8
	Applied Learning	18	10.4
	Scaffolded Learning	16	9.2
Lesson Implementation	Strengths	23	13.3
	Challenges	55	31.8
KIBO Organization	Strengths	4	2.3
	Challenges	21	12.1

Note. Asterisks are used to mark double-coded statements, which were already counted in the sum of code frequencies as part of other codes.

Knowledge

Teachers reported on the kinds of knowledge gains and challenges they experienced while participating in the CAL-KIBO program. Because this was the first coding experience for these teachers, the predominant form of knowledge teachers reflected upon was knowledge in the specific content area of coding and robotics. Teachers discussed gains in their KIBO knowledge and knowledge of screen-free coding. They also shared many challenges around mastering specific KIBO concepts (e.g., sound, light and distance sensors and repeat loops) and their overall lack of coding and robotics knowledge. The other form of knowledge reported by teachers was teaching knowledge, specifically regarding classroom management. Teachers expressed that knowledge of classroom management was essential for keeping students on track and maintaining proper care of KIBO materials. Although teachers also discussed various knowledge gains and challenges for their students (e.g., improved sequencing and use of details, learning how to work together and share ideas, etc.), these codes did not pertain to teacher knowledge and thus were not considered for this study.

Attitudes

Teachers expressed positive and negative views about various aspects of the CAL-KIBO program. These aspects were grouped under the following categories: attitudes around KIBO, attitudes around teaching coding and robotics, and attitudes around integrating coding with literacy and other content areas. Positive attitudes around KIBO included being excited about specific KIBO blocks and modules and feeling that KIBO was a hands-on, engaging, and user-friendly tool to introduce to young children. Teachers expressed negative views about the difficulty of moving KIBO kits in and out of the classroom. In addition, the excitement and novelty of KIBO diminished for some students after a few lessons, making it more challenging for teachers to maintain high student engagement.

Teachers also expressed positive and negative attitudes about teaching coding and robotics. Positive attitudes consisted of their own personal excitement about teaching, being open to learning, noticing students' excitement around learning coding, being receptive to bringing educational technology in their classroom, and feeling prepared and confident about teaching. On the other hand, teachers expressed feeling nervous about their lack of experience, being able to answer students' questions, and having students work together. Some teachers also expressed resistance to using educational technology in their classroom, making comments such as "I'm not technically savvy" or "technology is not my thing". Teachers also expressed general concerns about curriculum alignment and whether the CAL-KIBO program met the district's strict benchmarks for teachers and students.

Finally, teachers expressed positive and negative attitudes about integrating coding with literacy and other content areas. Due to the curriculum's emphasis on the parallels between CS and literacy, these connections were prioritized in the teacher interviews. Some teachers expressed positive views about integrating coding and literacy and chose to implement CAL-KIBO lessons as part of their reading and writing instructional time. Other teachers expressed skepticism and resistance to integrating coding and literacy, feeling that the connections were too forced, and instead implemented lessons during a non-literacy block. Some teachers felt stronger about integrating coding with other content areas, such as using KIBO to teach life cycles in a science unit or to teach patterns in a math unit. Overall, across the entire sample and all interview timepoints, teachers expressed more positive than negative attitudes about KIBO, teaching coding and robotics, and integrating coding with literacy and other content areas.

Beliefs

Three categories of teacher beliefs were identified: beliefs about coding, beliefs about coding and inclusion, and beliefs about teaching and learning in general. First, teachers shared beliefs about the overall benefits of coding, expressing views that coding supported students' creativity, learning in other areas, independence, new ways of thinking, and problem-solving skills, and that coding was an important 21st century skill for students. Teachers also shared beliefs specifically around the benefits of coding for young children, for instance, that young children have the capacity to acquire new knowledge more quickly, and that it was important for students to start young.

The second category of teacher beliefs focused on coding and inclusion. Some beliefs centered around expanding access to coding education, for instance, when teachers expressed beliefs that coding should be part of the regular curriculum or that coding is for all students, regardless of background or ability level. Other beliefs centered around specific groups of students. For example, teachers reflected on how their inclusion students engaged with KIBO and the lesson activities, noting that some students thrived and performed better than their peers, whereas other inclusion students needed additional support. One teacher described one of her students with high KIBO engagement: "I was very shocked because he's normally not as verbal and knowledgeable about a lot of things… It seemed like already had a good amount of pre-knowledge about robotics and coding and stuff like that." On the other hand, another teacher noted, "I have one student that is very, very... She's very, very behind. So she enjoys KIBO but she doesn't understand it as much as the others," and went on to describe the kinds of supports she provided the student with KIBO activities as well as in other subject areas. Gender dynamics were also brought to light, with teachers stating their belief that it was important to disrupt gender stereotypes around coding and sharing their excitement around female students excelling with KIBO.

The final category of teacher beliefs focused on their overall principles of education. Two major codes were identified: beliefs about learning as an ongoing, iterative process, and beliefs about having fun while learning. Both categories consisted of codes that represented teachers' broader views of education and their roles as early elementary educators. For instance, teachers believed it was important to model a lifelong learning mindset for students and demonstrate a growth mindset when faced with programming challenges. Teachers also held the belief that learning can and should be fun, and that students should experience joy in the learning process.

Behaviors

Teacher behaviors were identified under the following categories: pedagogical strategies, lesson implementation, and KIBO organization. Pedagogical strategies consisted of various approaches teachers used to facilitate students' learning. One set of pedagogical strategies was collaborative learning, which included strategies like sharing programs with peers, talking about programming challenges in large and small group discussions, and enabling students to work together in groups. Another set of pedagogical strategies was classified as applied learning because it consisted of strategies teachers used to make authentic connections to what students already knew. For instance, teachers discussed how they activated students' prior knowledge on a topic or made real-world connections to support students with mastering lesson objectives. The third set of pedagogical strategies was scaffolded learning, or breaking down complex tasks to better facilitate students' understanding. Examples of scaffolded learning included teachers helping students debug their programs and emphasizing to students that they should plan out their codes in their design journals before programming their robots.

Another component of teacher behaviors was how they implemented the CAL-KIBO lessons. These codes were classified as strengths or challenges, depending on how teachers reflected on their lesson implementation. Strengths included having designated days and times so that the lessons became part of the weekly routine, breaking up lessons into small chunks throughout the day, and setting a school schedule where teachers implement lessons on different days. This latter behavior enabled teachers to maximize the number of KIBO kits available for their classroom and made it possible for the district coordinator and ITRT to visit all classrooms at their designated times and offer support as needed. As such, one of the reported challenges was implementing lessons on the same days as fellow teachers. In addition, one school had an open layout, where four different classrooms occupied the same large learning space. A reported challenge from teachers at this school was that implementing lessons in this large space was disruptive to the other classrooms. A major challenge reported by all teachers regarding lesson implementation was time. Teachers reported general concerns about time constraints, as well as specific time constraints with respect to lesson preparation, lesson activities, and cleanup. Teachers also expressed that they sometimes shortened lessons to focus on other content instruction.

Teacher behaviors also consisted of how they organized the KIBO robotics kits in their classrooms and schools. One identified strength was having a central storage space for the school, so that teachers could easily access the KIBO kits on the days they were implementing lessons. For instance, School S had the KIBO kits stored in the library. Other schools chose to split up the KIBO kits, so that each teacher had several kits in their own classroom. Although this arrangement made it easier for teachers to manage their own materials, it limited students' ability to work in small groups and have substantial hands-on time with KIBO. On the other hand, teachers who shared kits with other classrooms expressed various challenges with keeping the kits organized with all their necessary pieces and putting the kits away properly after each use. Across the full sample and interview timepoints, teachers expressed more challenges than successes with respect to lesson implementation and KIBO organization.

Support

Teachers identified various sources of support that impacted their experiences with the CAL-KIBO training and curriculum implementation. Two major categories of support were identified: human supports and material supports. Human supports, as the name suggests, included any individual or groups of individuals that the teacher specifically pointed out as a source of assistance. These individuals included the school team (e.g., fellow grade level teachers, school principal, and other school personnel such as the librarian or paraprofessional), researcher-coach (i.e., a member of the research team who facilitated the professional development training, sent weekly email blasts, visited classrooms, etc.), ITRT (instructional technology resource teacher assigned to each school), district coordinator (responsible for on-site support and coordination for all teachers), and children (e.g., students in the teacher's classroom, or the teacher's own children). Code frequencies revealed that the school team and researcher-coach were the two most common sources of human supports identified by teachers.

Any non-human sources of support were categorized as material supports. This sub-category included visuals, such as anchor charts, lesson slide decks, and video tutorials on KIBO or the CAL-KIBO lessons. In addition, this sub-category also included curriculum materials, such as the printed curriculum binder and any online curriculum resources. Teachers commented on the overall format and structure of the curriculum, as well as specific lesson activities. One teacher asked for arts and crafts supplies to be included with the KIBO robotics kit; although this was included in the codebook under material supports, it was only mentioned once in the entire dataset.

Once initial codes were generated for these statements, any statement that indicated a teacher's explicitly positive or negative view on the quality of support was given a secondary code. For example, a statement like "[The district coordinator] was wonderful and she would always check in on the kids with any kind of concerns or questions that we had" indicated that the teacher had a positive perception of the support received by the district coordinator. Conversely, a teacher who commented on her fellow grade level teacher, saying "She said she just didn't wanna talk about it 'cause she didn't have time," was marked as negative perception of support from the school team. Statements that were neither explicitly positive nor negative did not receive a secondary code.

Changes Over Time

After all codes were identified, the subsequent steps of thematic analysis involved grouping the codes into themes and looking for patterns across timepoints, teachers, and schools. Although not every teacher participated in every interview, it was useful to examine the holistic distribution of codes across the five interview timepoints: pre-training (see Figure 6), post-training (see Figure 7), pre-curriculum (see Figure 8), mid-curriculum (see Figure 9), and post-curriculum (see Figure 10).

As shown by the dominance of positive attitude codes in Figure 6, at the beginning of the study prior to the CAL-KIBO training, teachers expressed very positive views about introducing their young students to coding and robotics. Teachers' positive attitudes were connected to their instructional behaviors and beliefs. For instance, their feeling of excitement around teaching coding stemmed from the belief that learning to code had many benefits for young children and would enable children to apply their learning to the outside world of computers and technology. However, these teachers acknowledged their own lack of knowledge and expressed immediate concern about fitting the CAL-KIBO lessons into their weekly plans. At this early stage of the program, the two sources of support teachers acknowledged were ITRTs and children. ITRTs were responsible for supporting teachers with all technology-related needs, so it was unsurprising to find that teachers identified them as a key source of support. However, the identification of students (or in one case, the teacher's own children) as knowledgeable individuals around the subject of coding and robotics was surprising. This finding exemplified teachers' awareness that students may have more expertise in this area and that this program would offer teachers the opportunity to learn alongside their students.

Figure 6

Pre-Training Visual Map

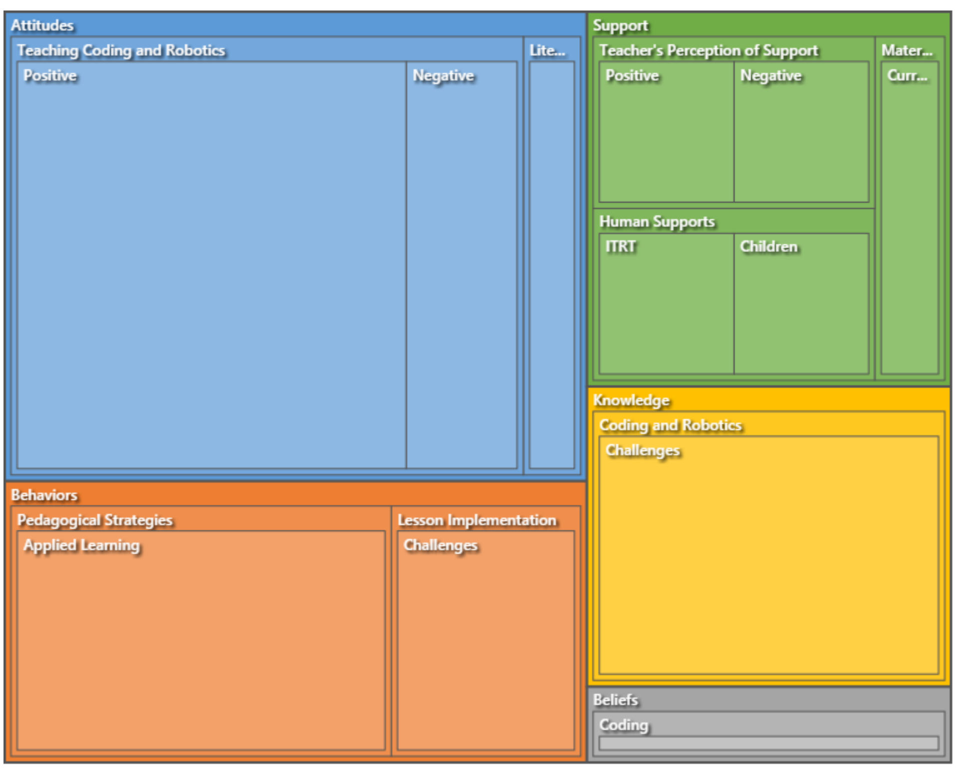
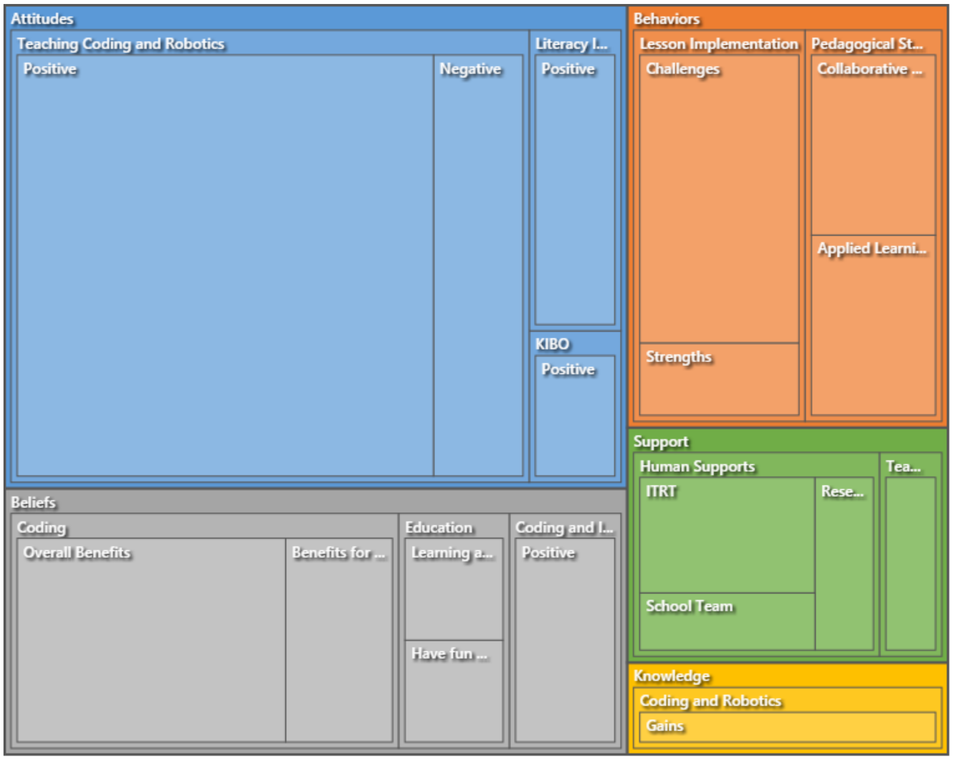


Figure 7

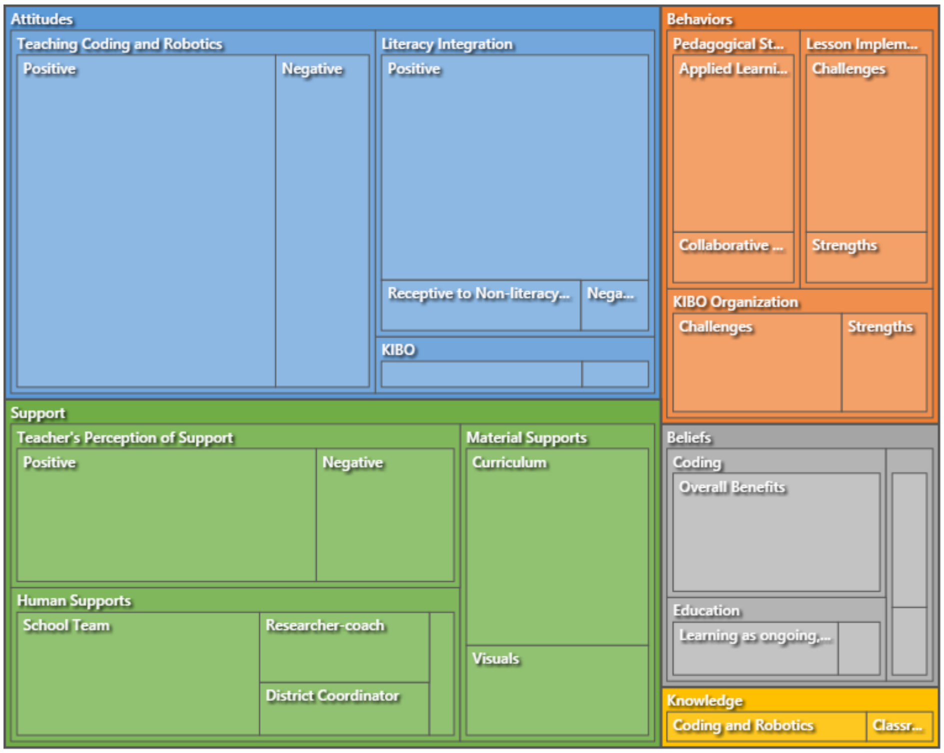
Post-Training Visual Map



Post-training interviews were conducted with teachers either on the same day or within two weeks of the professional development training. As displayed in Figure 7, teachers continued to express predominantly positive attitudes towards teaching coding and robotics, and it was evident that the training was successful in supporting teachers' coding knowledge and beliefs. Although teachers also continued to express concerns about lesson implementation, a few teachers had begun discussing plans with their colleagues and felt comfortable with integrating CAL-KIBO activities into their literacy instruction time. In addition, teachers began to see collaborative learning as an effective pedagogical strategy to support students' KIBO experiences. In terms of support, teachers continued to view their ITRTs as essential to successful implementation, but they also now acknowledged the research team and their own colleagues and administrators as additional resources.

Figure 8

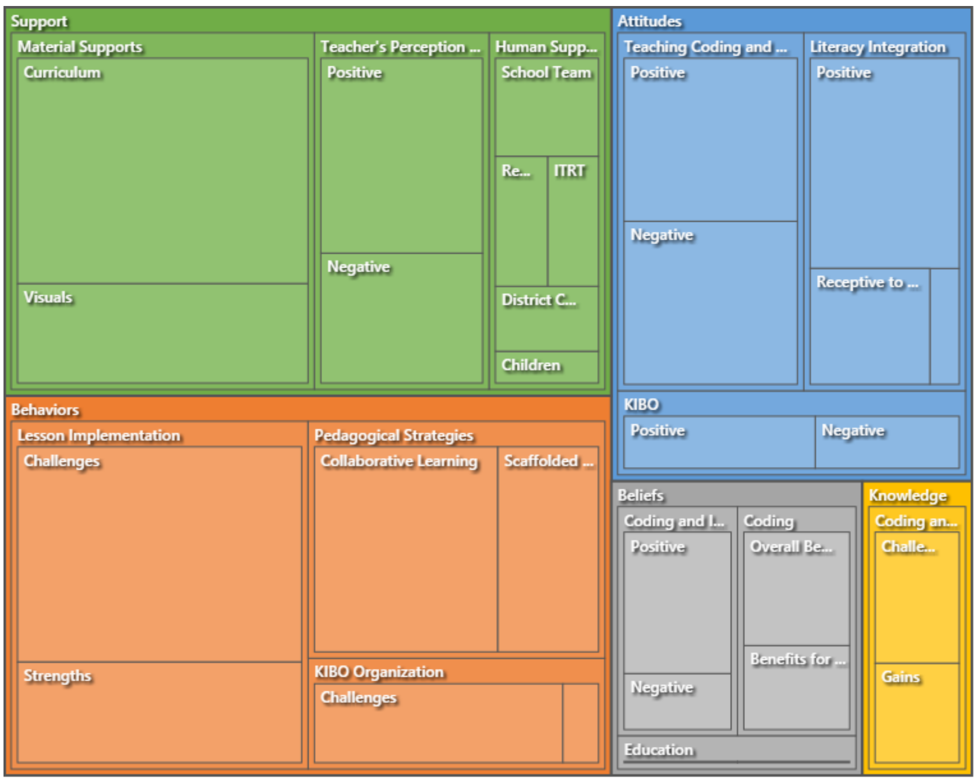
Pre-Curriculum Visual Map



A common pattern in teachers' pre-curriculum interviews was teachers' prioritized focus on the practical and logistical aspects of curriculum implementation, which was further evidenced in the distributions of codes at this timepoint (see Figure 8). Teachers described taking time to individually read through the first several lessons and having key conversations with their school team about implementation logistics. These conversations included answering questions such as how many KIBO kits teachers will have, when lessons will be implemented, and how one teacher's implementation schedule might impact the schedules of other teachers. Teachers expressed predominantly positive views about literacy integration and the kinds of support they needed and received, primarily from the school team, research team, and district coordinator (who visited each school to drop off printed curriculum binders for teachers). However, teachers continued to express challenges, specifically around time and KIBO organization.

Figure 9

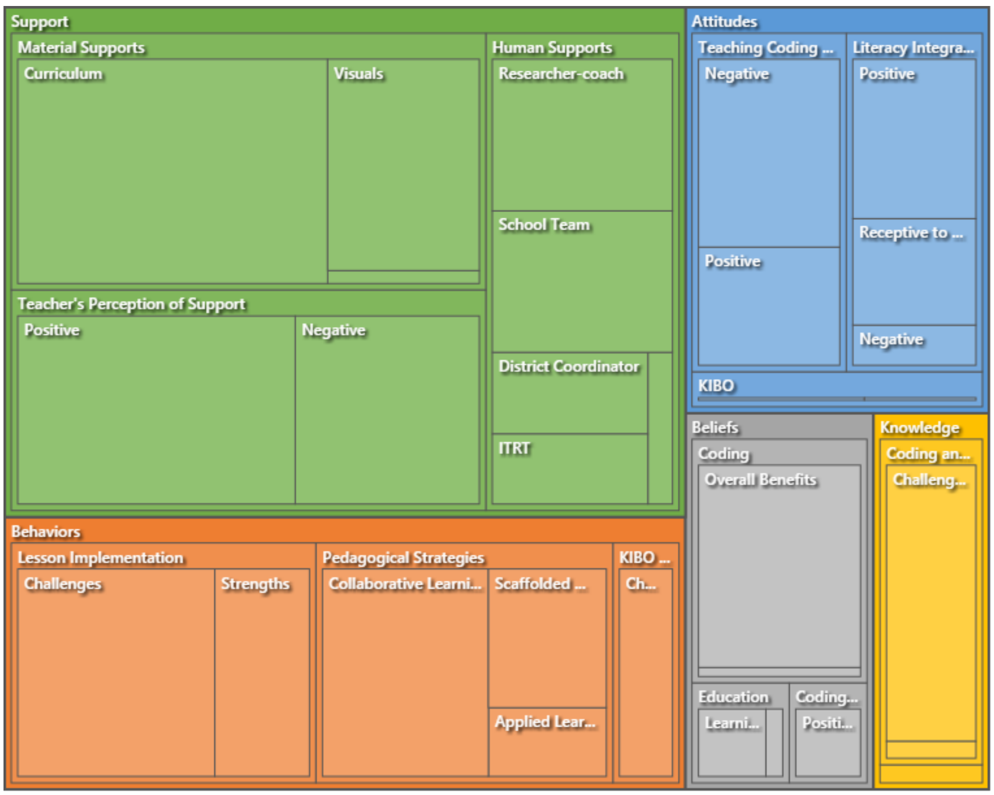
Mid-Curriculum Visual Map



By the middle of the curriculum implementation phase of the study (see Figure 9), teachers had established routines and implementation schedules, and their interviews focused mostly on the ways in which they prepared for and executed lessons. Teachers described relying on the written lesson plans but desired more visuals and child-friendly scripts that would make it easier to prepare for lessons. As teachers introduced more advanced KIBO blocks and modules (which required setting and cleaning up additional materials), lack of time became a more substantially prominent code that impacted teachers' instructional behaviors and attitudes towards KIBO and towards teaching coding.

Figure 10

Post-Curriculum Visual Map



By the end of curriculum implementation, all teachers shared both positive and negative aspects of their experiences and attitudes. Teachers faced additional challenges with lesson implementation and concerns about time constraints, which contributed to increased negative attitudes about teaching coding. The research-coaching team and school team were the two most frequently mentioned sources of support. Although teachers continued to recognize the overarching benefits of their students learning to code, they noted from observing students that one of the biggest benefits of the CAL-KIBO program was students learning how to collaborate and work together on their final projects.

When investigating patterns across teachers and schools, it became evident that some teachers shared similarly positive experiences and attitudes, whereas other teachers shared more moderate or even negative experiences and attitudes. Further investigation into these differences led to a deeper understanding of individual and contextual factors that impacted teachers' experiences and attitudes, which then became the focus of the second research question.

Research Question Two: To what extent did teachers' experiences and attitudes vary, and what kinds of individual and contextual factors might explain these differences?

Three groups of teachers were identified, each of which shared similar experiences and attitudes (see Table 7). The first group of teachers, who all had over ten years of teaching experience, voiced substantial challenges with lesson implementation. These teachers either expressed predominantly negative attitudes towards teaching coding and robotics, or negative perceptions of support. This group was classified as "Resistant but Committed" and represented by the quote *"I do what I need to do…but it's not my thing"*. The second group, with an intermediate level of teaching experience, also expressed challenges with lesson implementation but exhibited more moderate attitudes and perceptions of support. This second group was classified as "Eager but Overwhelmed" and summarized by the quote *"I totally support the idea, but it's really overwhelming"*. The third group, who had a mixed range of teaching experience, exhibited positive attitudes, perceptions of support, and beliefs about coding, in addition to sharing many highlights of their overall program experience. This group of teachers was classified as "Receptive and Enthusiastic" and summed up by the quote *"I think it's crucial. We have to have it and we need it implemented at the youngest, earliest age that we can"*. These three groups of teachers are presented in these analyses as themes because they capture the overall essence of teachers' reported experiences and attitudes. The following sections describe these themes further, along with illustrative case study examples. Please note that names of case studies have been changed to preserve the anonymity of participants.

Table 7

Overview of Teacher Types

Teacher Type	Count (*n*)	Schools Represented (Count)	Code Category		
			Perceptions of Support	Attitudes around Teaching Coding and Robotics	Challenges with Lesson Implementation
Type 1: Resistant but Committed	5	School T (2) School S (2) School G (1)	Mostly negative	Mostly negative	Many
Type 2: Eager but Overwhelmed	3	School C (1) School F (1) School B (1)	Moderate	Moderate	Some
Type 3: Receptive and Enthusiastic	5	School S (2) School B (3)	Mostly positive	Mostly positive	Time-based

Type 1: Resistant but Committed

Five of the 15 teachers stood out as having particularly negative experiences and attitudes around the CAL-KIBO program. The most significant and common concern were their challenges with lesson implementation, especially with time. Teachers struggled with finding time to implement lessons and expressed needing more time to prepare for lessons, clean up lesson materials, and focus on other content such as discrete literacy skills and mathematics concepts. When asked about supports teachers found most and least helpful, teachers expressed dissatisfaction around the availability of human supports. One teacher, for example, commented in her post-training interview: “how come nobody's coming in to provide that first day or second day training to our class, while we're there of course, to provide the first two days of training with the children?” Upon further discussion with this teacher, it was evident that her desire to co-teach lessons with a curriculum expert stemmed from the teacher’s nervousness about answering students’ questions and feeling ill-equipped and unprepared to implement the CAL-KIBO curriculum. This example illustrated a possible relationship between teacher attitudes towards coding and their perceptions of support.

When examining these teachers’ background characteristics, it was interesting to note that all five teachers reported having over ten years of teaching experience. In fact, for one of the teachers, this was her final year before retirement. This finding aligns with Papadakis and colleagues (2021) who found an inverse relationship between educators’ teaching experience and positive attitudes towards educational robotics. A possible explanation for this finding may be that teachers with many years of classroom teaching experience are less inclined to adapt to new instructional tools and curriculum. Strawhacker et al. (2017) found that teachers who exhibited flexibility with lesson planning and comfortability with technological content were the most successful in promoting students’ coding knowledge. In this study, these five teachers consistently acknowledged the difficulty of fitting in CAL-KIBO lessons with their existing curriculum and expressed feelings of anxiety and nervousness about their lack of comfort around technology.

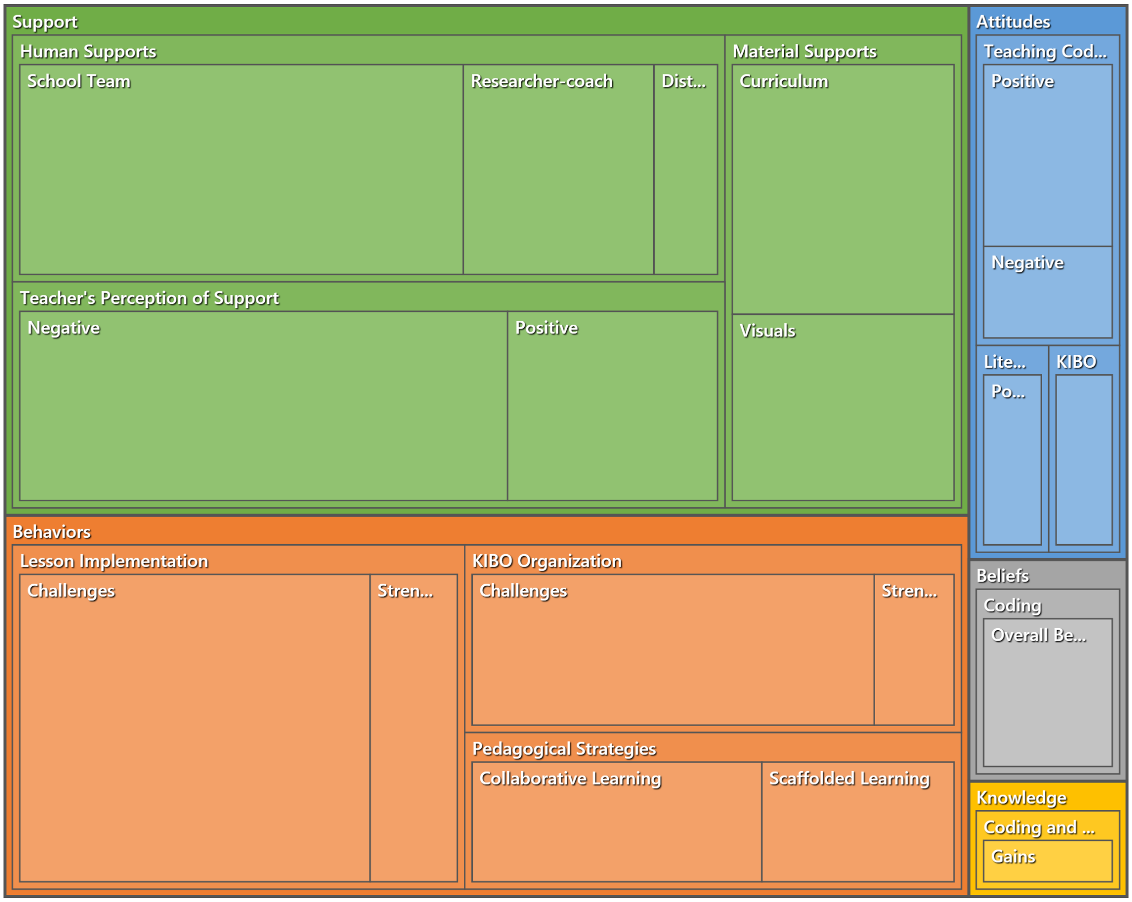
Case Study Example. Ms. Cooley is a White, female teacher at School G with over 15 years of classroom teaching experience. As evident from her visual map (see Figure 11), Ms.

Cooley experienced significant challenges with lesson implementation, KIBO organization, and support resources. In her interviews, she expanded on the lack of support from her school team, saying, “[District coordinator] has been wonderful as far as copying stuff for us and stuff like that. I guess just more what I'd need to do instead of look at it over the weekend. I kinda need to be with my colleagues, like the librarian so we can bounce ideas off of each other. 'Cause I did feel like I was alone and I don't know, there's nothing wrong with the other teacher, she just does things differently and didn't want to talk about it.” This quote exemplified Ms. Cooley’s desire to collaborate with her colleagues and lean on them for support. However, knowing this need could not be met, she remained committed to the experience and decided to plan independently, which contributed to her feeling of loneliness and increased concern about time.

In addition to lack of support, another contextual factor that impacted Ms. Cooley’s experiences and attitudes was School G’s open classroom layout. In an open classroom layout, multiple classrooms (often mixed grades) share a single large space with limited barriers and walls, which is intended to promote small-group and individual instruction and encourage collaboration and active learning. Although there are many benefits to an open classroom layout, in this context, the layout proved to be a challenge because not every classroom in the large space participated in the CAL-KIBO program. Ms. Cooley described in one of her interviews: “we have four classes in one room. And so on Kibo days, the other classes were just... We couldn't do anything about it; it was louder. It was distracting to the other teachers as well. But that's due to how our school is structured.” This contextual factor contributed to Ms. Cooley’s reported challenges around lesson implementation and her ability to use collaborative learning effectively as a pedagogical strategy to support students’ learning with KIBO.

Figure 11

Visual Map for Case Study Teacher Ms. Cooley



Type 2: Eager but Overwhelmed

Three teachers were characterized as having more moderate experiences and attitudes, meaning that they similarly expressed both negative and positive views. For example, these teachers were excited about bringing something new to their students, something that they acknowledged their students would really enjoy and benefit from. However, they also admitted to having concerns about curriculum alignment and about time taken away from other content areas. This balanced perspective of acknowledging both pros and cons were emulated in their reflections of the CAL-KIBO curriculum. Teachers appreciated the written curriculum and found the lesson plans to be a helpful teaching guide. They also offered feedback about incorporating additional visuals, which would make it easier for teachers to prepare for lessons. One teacher even made her own slide decks to accompany each lesson and shared this resource with fellow teachers, demonstrating the teacher's eagerness to enhance students' learning experiences and her willingness to be a human support resource for others.

Like the other five teachers, the three teachers grouped into this category faced challenges with lesson implementation. In addition to time constraints, one challenge reported by a teacher was having everyone at her school implement the CAL-KIBO lessons on the same days and times. The teacher described the dilemma in her post-curriculum interview: "I think it may have just been a problem with the way that our school set it up… [the district coordinator] couldn't really split herself between four classes. And then if we hadn't been doing it all at the same time, then we could have used all the KIBOs". This contextual factor of school implementation logistics impacted this teacher's ability to utilize human supports and maximize the number of KIBO robotics kits available for students to work in small groups, which likely influenced the teacher's overall experience and attitude towards teaching coding and robotics.

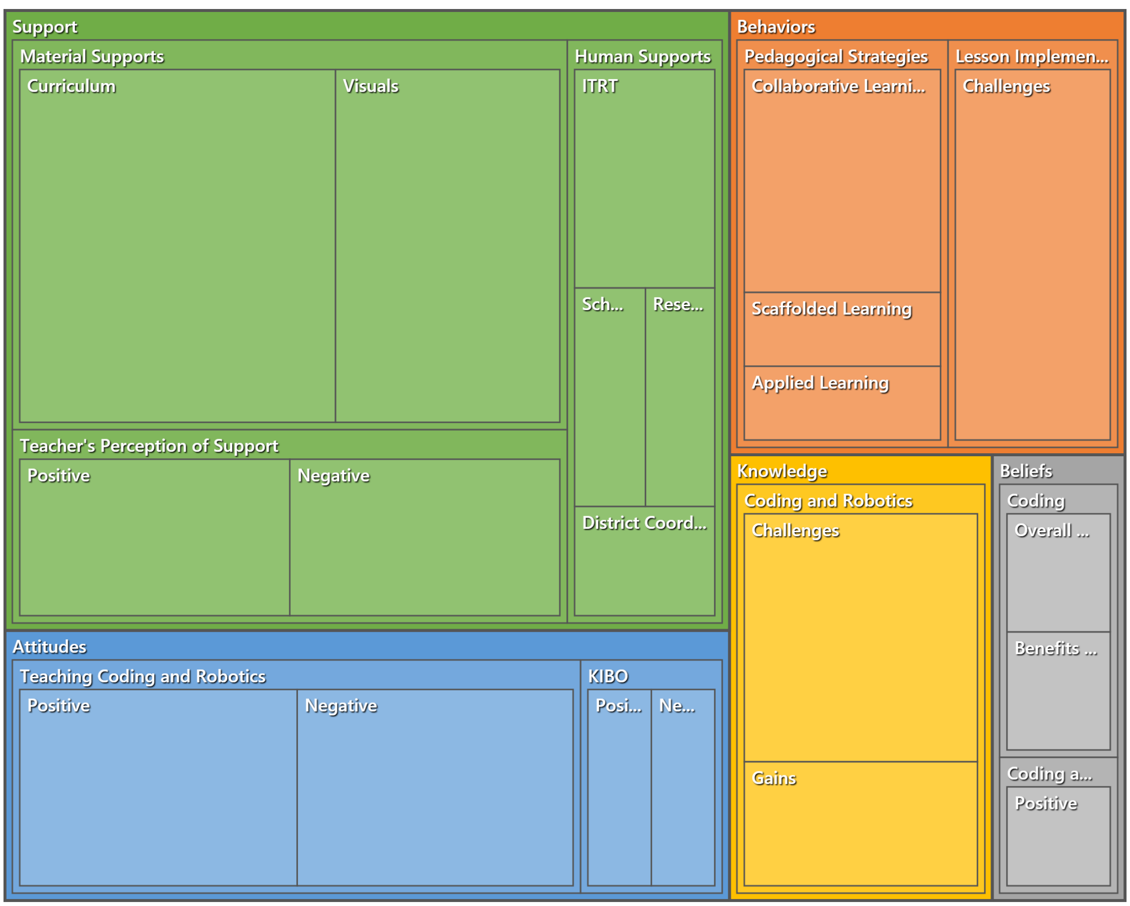
All three teachers had an intermediate level of teaching expertise. One teacher had taught for 4-5 years, and the other two teachers had 6-10 years of teaching experience. Prior work has indicated that teachers with an intermediate level of teaching expertise have the most optimal experiences introducing new curricular programs. The rationale is that novice teachers are likely still honing effective pedagogical practices, classroom management skills, and learning how to be "good" teachers. Veteran teachers, on the other hand, are more likely to exhibit strong pedagogical practices but are often more resistant to change. Thus, it may be that in this study, these teachers with intermediate teaching expertise were able to critically reflect on the successes and challenges of their experiences and offer both positive and negative perspectives.

Case Study Example. Ms. Potts is a White, female teacher at School B and at the time of the study, had 6-10 years of classroom teaching experience. Prior to beginning the curriculum with students, Ms. Potts exhibited moderate attitudes towards implementing the CAL-KIBO lessons (see Figure 12). She described her feelings in the interview, “I think a little bit of mixed emotions with excitement for my kids to be able to start with the training and then maybe a little bit of anxiety over how I'm going to start it.” Her anxiety about starting lessons stemmed from the school’s plan to take the entire second grade class to an off-campus camp for ten days, which interfered with the tentative start date of the CAL-KIBO curriculum. Thus, Ms. Potts’ concerns regarding lesson implementation were influenced by other school priorities and the inability of teachers to transport materials to the camp site, figure out implementation logistics, and accordingly prepare for lessons.

After Ms. Potts began implementing the curriculum (when the camp was over), she continued to express both positive and negative views, particularly with respect to support resources. On one hand, Ms. Potts felt that human supports were helpful but not necessary. She commented in an interview, “I was pretty comfortable and confident in my abilities to implement the lessons, so I didn't really need anybody to come in and give me support. But then also, [district coordinator], when she was in one day, I had... There was a teacher out, so I actually had seven extra kids in the class and I was trying to teach KIBO. So, she was really helpful in crowd control and also taking the kids and separating them and helping teach a lesson to them while I taught another group.” This quote highlights Ms. Potts’ ability to effectively implement lessons independently unless external circumstances (such as a teacher being out) necessitated having additional hands-on assistance. On the other hand, Ms. Potts felt the material supports were inadequate for her teaching needs. She described in an interview, “I'd say from Lesson One to Lesson Six, I actually created smart board slides, because there was too much too much information in the lesson plan for me to be able to effectively teach. Because I've gotta be engaged in a lesson, so that way they're engaged, and if my head's always in the notebook, then I'm not engaged.” Ms. Potts’ decision to create additional visual aids to enhance student engagement and learning in her classroom illustrates a connection between teacher support and pedagogical strategies.

Figure 12

Visual Map for Case Study Teacher Ms. Potts



Type 3: Receptive and Enthusiastic

Five teachers shared predominantly positive reviews over the course of their participation in the CAL-KIBO program. These teachers varied in their years of teaching experience: two teachers with 0-3 years, two teachers with 6-10 years, and one teacher with over 15 years of teaching experience. This finding suggests that there is not necessarily a definite relationship between teaching experience and attitudes towards coding. In this case, the veteran teacher had a son who was a professional in the computing field and consistently referenced him during interviews. The teacher even described talking about her son with students to spur their interest about professional CS careers. This example goes to show how a teacher's individual background can influence how they approach CS teaching and learning.

Another commonality among these teachers was their beliefs about the overall benefits of coding and about general principles of education. It was evident that teachers believed that learning to code supported students in many ways other than just learning coding. For instance, teachers pointed out how students' participation in the CAL-KIBO curriculum supported creative thinking, particularly with the open-ended writing prompts in the design journals. One teacher noted that students were becoming "better problem solvers in all aspects of the day", and that was a direct result of students' engagement with KIBO, as well as the various opportunities students had to debug and troubleshoot errors. When asked how the CAL-KIBO curriculum fit into the rest of their classroom activities, several teachers pointed out general principles of education that guided their teaching philosophies. For example, multiple teachers reflected on learning as an ongoing, lifelong activity and used this experience to demonstrate that teachers were also learners. The iterative design process, compared against the writing process, was another salient connection that students understood and used to practice planning, revising, and sharing their ideas and KIBO robotics creations.

Although teachers mentioned many strengths of lesson implementation, these teachers also did express challenges. Time was again a major concern. To address this issue, some teachers extended single-day lessons over multiple days, whereas others shortened lesson activities to focus on other content areas. KIBO organization was another key challenge for teachers, especially with limited time allocated for clean-up. To help with this issue, some teachers described using the student job cards provided in the CAL-KIBO curriculum that enabled students to take on specific responsibilities such as cleaning up or scanning the KIBO blocks. These job cards also served as a useful collaborative learning tool, which was the pedagogical strategy most frequently identified by these five teachers.

Finally, these teachers demonstrated positive perceptions of support. Four of the five teachers highlighted the school team (i.e., fellow colleagues, principal, etc.) as an important human support resource. For instance, one teacher described the relatively easy process of sorting out implementation logistics with her school team:

> *Teacher:* We did work it out with our principal so that every second grade classroom gets all 10 for an hour, twice a week… Two of us are gonna change our schedules around. So I think that's gonna work out really great for the kids to be able to work in groups of two. And I think I'll have one group of three, but yeah, I'm looking forward to it.
>
> *Interviewer:* That's great. Was it hard to figure out the schedule or to get the principal buy-in for that? What was your process for that?

Teacher: No, it was actually way easier than I actually thought it was gonna be. I thought that she was gonna put up a big old fight. And we told her... She sat us down and she asked us what we thought about it, and we told her, and she 100% agreed with us. Things don't usually go that way. So it was very exciting when she said, "Yeah."

From this interview excerpt, it was evident that new programs are not always met with positivity, particularly when administrators may have competing priorities. However, this example shows how teachers and administrators shared the common goal of integrating the CAL-KIBO program into the curriculum and were able to collaborate effectively to sort out logistics as a team. This finding speaks to the positive staff culture and shared vision at this school for promoting CS teaching and learning, embodying what other researchers (e.g., EdTech Evidence Exchange, 2021; Ertmer & Ottenbreit-Leftwich, 2010) have identified as key variables for effective technology integration in schools.

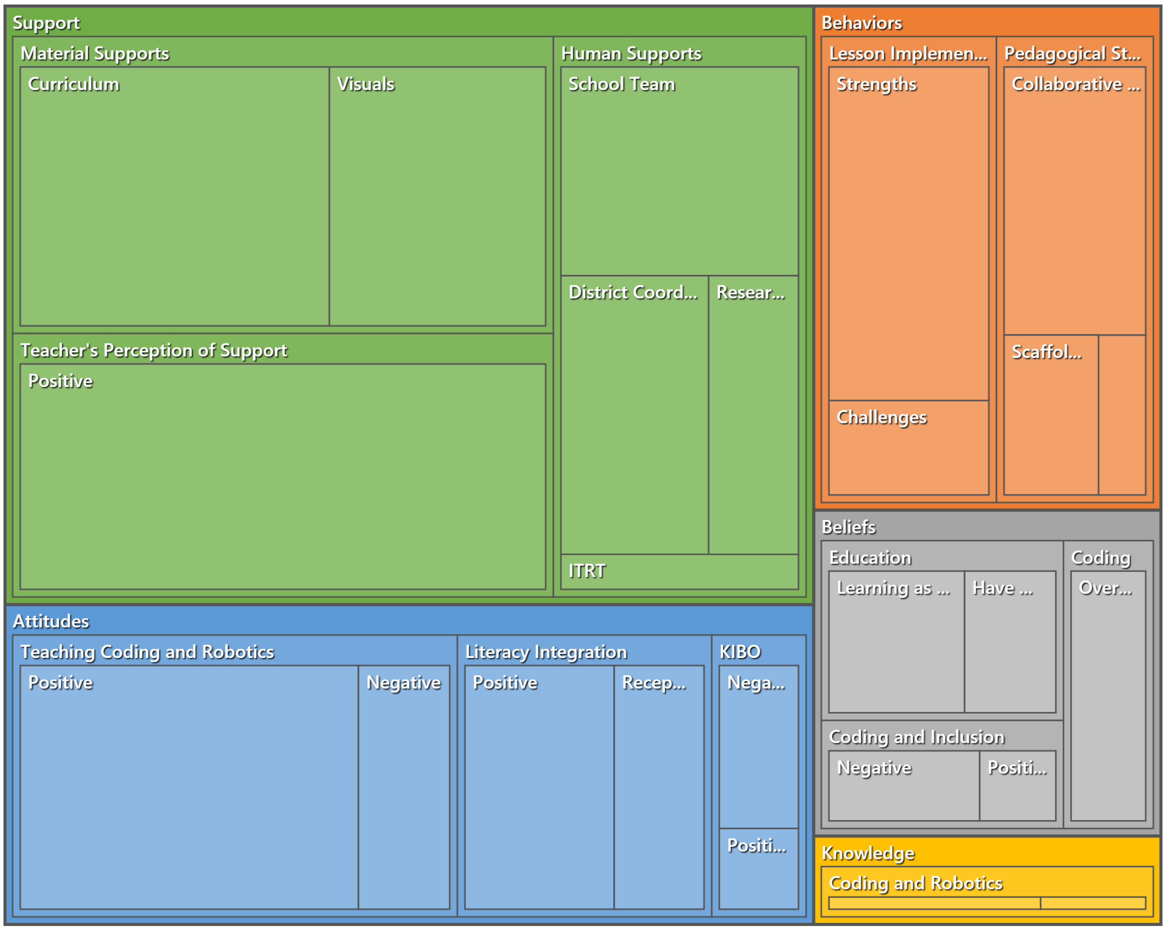
Case Study Example. Ms. Bowman is a Black, female teacher at School S and at the time of the study, had 6-10 years of teaching experience. At the training, Ms. Bowman exhibited an initial lack of confidence around technology usage, saying “I was very reluctant to come candid because I'm just not so technically savvy.” However, she was eager to learn and thoroughly enjoyed the professional development training. In preparing to implement lessons in her classroom, Ms. Bowman acknowledged her nervousness about teaching something new but felt adequately supported by the human and material supports. She describes in an interview, “I would say I'm a little nervous just about something new, but I get nervous about doing new things anyway. But I feel pretty confident in everything, just because we've had a lot of support, the training, and then [district coordinator], we get to talk to her a lot.” Her acknowledgement of multiple avenues of support and the ease of access to these resources contributed to her increasingly positive attitudes around teaching coding.

By the end of the curriculum, Ms. Bowman expressed high enthusiasm about all aspects of her CAL-KIBO program experience, with her primary concern being that she had wanted more time for the final project lessons. Over the course of the program, she seemed to develop strong beliefs about the benefits of coding for young children and how learning to program could support children's learning in other areas. For example, during her post-curriculum interview, Ms. Bowman reflected on her views on coding-literacy integration, saying, "the editing piece to me was very strong, going back and fixing things and not getting frustrated when you face challenges and really looking at it as, 'Oh, something else for me to figure out.' And just becoming a little detective, investigators they refer to themselves as programmers and stuff". This quote demonstrates how Ms. Bowman viewed the experience as transformative for her students' STEM identities while also acknowledging how her students engaged in learning important human virtues such as perseverance and optimism (Bers, 2022).

When looking holistically at the distribution of codes for Ms. Bowman (see Figure 13), it is evident that she exhibited predominantly positive experiences and attitudes. However, what is not shown in this map is the growth she experienced over the course of the program. In her final interview, she summarized, "When you're a teacher, it's always something new. Someone always wants to tell us something new that we need to do, and I think when we first found out that we would be doing coding, it sounded intimidating, being that we didn't know anything about it, and then I think the training kind of eased our minds a little bit, and then actually doing it and seeing how much the kids loved it and the benefit that it had." This reflection epitomizes how her experiences and attitudes shifted over the course of the program, and specifically how the training (support) and curriculum implementation (beliefs and behaviors) contributed positively to these changes.

Figure 13

Visual Map for Case Study Teacher Ms. Bowman



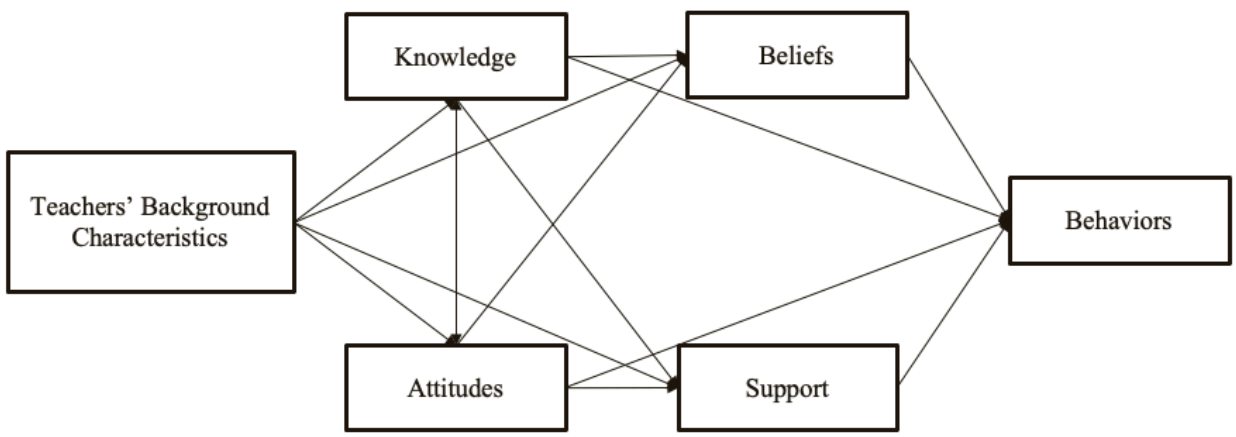
Summary of Findings

The first research question of this study aimed to explore second grade teachers' reported experiences and attitudes over the course of their participation in the CAL-KIBO program. Using Ernest's (1989) theoretical framework as a starting point for identifying codes and themes, I identified five distinct but related categories that encapsulated teachers' experiences and attitudes. Teacher knowledge was comprised of knowledge of coding and robotics, as well as knowledge of classroom management. Teacher attitudes included attitudes towards coding and robotics, as well as attitudes towards *teaching* coding and robotics. Teacher beliefs encompassed beliefs about coding and robotics, beliefs about coding for young children, and beliefs regarding overall principles of education. Altogether, these three factors influenced teachers' instructional practices and behaviors regarding lesson implementation, KIBO organization, and pedagogical strategies.

Whereas Ernest's (1989) model included teacher knowledge, attitudes, beliefs, and behaviors, he did not include support as a key factor that influenced instructional practices. However, findings from this study indicate that support (in the form of human and material resources) was a critical component of teachers' experiences and was related to other factors such as teacher knowledge, attitudes, and behaviors. For example, if a teacher was struggling to figure out how to work KIBO's sound sensor, she sought out help from a colleague or asked the research team for additional resources. If a teacher did not receive adequate support (e.g., the lesson plan did not contain enough visuals), then this lack of support contributed to the teacher's negative attitudes towards teaching coding. Conversely, the lack of support may have also motivated the teacher to create her own visuals, which impacted her instructional behaviors. These examples highlight how support should be an integral component of Ernest's (1989) theoretical framework. Although the nuances of these relationships should be examined further with larger datasets and additional qualitative and quantitative measures, a modified version of Ernest's framework based on the findings from this study is proposed in Figure 14. In this revised framework, teacher support is linked to background characteristics, knowledge, attitudes, and behaviors. In this study, there was no explicit evidence for a relationship between beliefs and support.

Figure 14

Relationships Among Teacher Knowledge, Attitudes, Beliefs, Instructional Behaviors, and Support



The purpose of the second research question was to expand on the first and further investigate the extent to which teachers' experiences and attitudes varied, and the kinds of individual and contextual factors that might explain those differences. Findings revealed three groups of teachers: 1) "resistant but committed" teachers who shared similarly negative experiences and attitudes, 2) "eager but overwhelmed" teachers who shared moderate views, and 3) "receptive and enthusiastic" teachers who shared similarly positive experiences and attitudes. Each teacher had individual and contextual factors that contributed to their unique views. Examples of such individual factors included teaching experience and personal connections to the CS field. In addition, examples of contextual factors included classroom layout and school culture. These findings are comparable to the existing literature on teaching coding in early elementary classrooms. For instance, teachers' lack of knowledge and need for support materials were identified as key challenges by Khanlari (2016) and Kong (2019). In addition, Burke and colleagues (2020) found a relationship between teachers' positive attitudes towards CS education and positive beliefs about the benefits of coding for students' lifelong learning skills. In this study, however, having positive beliefs was not always related to having positive attitudes. There were Type 1 teachers who exhibited strong beliefs about the benefits of coding, particularly for young children, but did not feel positively about their ability to teach coding effectively to their students. Type 3 teachers, on the other hand, exhibited both positive attitudes and beliefs, further emphasizing the need to look at the complexities of these relationships.

This dissertation aimed to fill an important gap in examining teachers' experiences and attitudes not only over the course of a professional development training, but also over the course of implementing a full-length robotics curriculum with students. Studies such as Ensign (2017) show that active participation in activities with students can promote teachers' confidence. In this study, findings indicated it was not always the case. Some teachers, despite reporting growth in their coding knowledge, still did not feel fully confident in their ability to teach coding. Other teachers, such as Ms. Bowman, reported a transformative change in their confidence and attitudes towards teaching coding. Ms. Bowman even explicitly credited part of this growth to the high engagement and learning she saw with her students. This finding illustrates the point that professional development without the opportunity to put learning into practice may not necessarily yield the most optimal outcomes for both teachers and students.

Chapter Six: Conclusion

The purpose of this study was to understand second grade teachers' experiences and attitudes around teaching coding and robotics in their early elementary classrooms. Teachers were introduced to the Coding as Another Language (CAL) pedagogical approach and the screen-free KIBO robotics platform in a professional development training. Subsequently, teachers implemented the CAL-KIBO curriculum in their classrooms approximately twice a week and were interviewed at various points before, during, and after the training and curriculum. Analyses of these semi-structured interviews indicated five key categories encompassing teachers' experiences and attitudes: teacher knowledge, attitudes, beliefs, behaviors, and support. Although each of the 15 teachers participated in a similar CAL-KIBO training and implemented the same CAL-KIBO curriculum, their experiences and attitudes around the whole program varied due to individual and contextual factors. Some teachers' reflections leaned towards a more negative viewpoint, whereas other teachers shared more moderate or mostly positive aspects of their experiences. Despite these differences, every teacher had pros and cons, which provided useful feedback for the research team to refine the CAL-KIBO training and curriculum for future teachers.

Recommendations for Practitioners

As evident from the findings from this work, each teacher has unique background characteristics that must be considered when determining how to best support the teacher in their endeavors to bring coding and robotics into their respective classroom. However, the following general recommendations (based on the three groups of teachers identified in this study) may prove useful for instructional coaches and other practitioners responsible for providing supports for early elementary teachers:

For “Resistant but Committed” teachers, designate a human resource for the teacher early in the process to assist the teacher with lesson preparation and implementation. Expert guidance will enable these teachers to feel more comfortable and confident about their own learning and their capacity to engage students in learning coding. Furthermore, building a strong foundation for the teacher-coach relationship will encourage the teacher to stay committed and to reach out for support whenever they experience challenges or frustrations.

For “Eager but Overwhelmed” teachers, human resources may be helpful, but coaches may not need to be as hands-on. Rather, these teachers will benefit when their most pressing challenges are addressed in a proactive manner. For instance, these teachers may be quick to recognize technological issues or the lack of specific curriculum resources. Even if teachers may be eager and capable of addressing these issues themselves, the support of instructional coaches to help address these issues alongside teachers may alleviate any additional stress placed upon teachers.

For “Receptive and Enthusiastic” teachers, check in with them on a regular basis to make sure the supports already in place are either maintained or lessened if the teacher no longer requires them. To enhance these teachers’ instructional practices, coaches might observe classrooms and provide resources on how to best support specific students or how to provide differentiated instruction so that all students feel appropriately challenged. These teachers may also be ready to develop their own lesson plans and materials, in which case they may benefit from seeing examples of other integrative curricula.

Implications for Research, Practice and Policy

The development of coding technologies for young children is relatively recent, as is the use of these technologies in early childhood classroom settings. Findings from this dissertation add to the existing literature on elementary CS education, which currently lacks the breadth of study present in higher grade levels. Furthermore, this dissertation utilizes a diverse sample of second grade educators from a large U.S. public school district. Unlike small-scale studies that are often conducted in private school settings or afterschool programs, this study was conducted in real-world classrooms and offers important insight into the nuances and complexities of introducing CS in public education contexts. Although the sample was not randomly selected and represents only a subset of a single school district, this work can still help inform future research studies examining the integration of coding and robotics in mainstream curriculum. Regarding implications for practice, with the expansion of computer science in primary and secondary education, there is an increasing need to train teachers in computing. Understanding early elementary teachers' views on coding and robotics education enables us to develop more effective and targeted professional development and curriculum resources.

The push to introduce coding and robotics in early elementary classrooms has been largely driven by national and international policies to create and implement CS education frameworks and standards. To support the implementation of these policies, millions of dollars are spent each year to support CS education efforts in the U.S. and globally. However, it is unclear how to best allocate these funds and develop effective policies and programs that improve CS experiences for teachers and students. Findings from this dissertation can offer policymakers additional guidance about how to make these decisions with the backing of research evidence and practitioner input.

Study Limitations

Sample

This study was limited by a small sample of 15 teachers from a single grade level from a single school district in the United States. Although this sample was appropriate for the type of qualitative analysis conducted, the sample size does limit the generalizability of findings. Furthermore, because the study was conducted in real-world educational settings, it was expected to have some level of attrition with teachers' interview participation over the course of the study. There were unexpected absences, teachers pulled in multiple different directions with other school initiatives, and personal issues that impacted teachers' ability to participate fully in the CAL-KIBO program and research. Although efforts were taken to minimize these issues and to schedule interviews around teachers' open planning times, not every teacher was interviewed at every timepoint. This limitation impacted the ability to conduct longitudinal analyses at the individual level. Future work should aim to replicate this study with a larger sample of teachers. Researchers might collaborate with school staff to maximize interview participation, so that possible changes in teachers' reflections before and after the CAL-KIBO program can be explored.

Curriculum Implementation

This study was the first multi-classroom implementation of the CAL-KIBO training and curriculum. Due to the exploratory nature of this pilot study, there was no pre-set standard for fidelity of implementation. Teachers were asked to collaborate with their school team to navigate implementation logistics with the backend support of the district coordinator and research team. Thus, implementation logistics were not standardized across schools. Furthermore, half the schools participated in the CAL-KIBO program in Wave 1 (November – March) and the other half participated in Wave 2 (March – June). The gap between the training and curriculum was several months for Wave 1 and only several weeks for Wave 2. Although findings did not indicate wave as a key factor impacting teachers' reported experiences, at least one teacher from Wave 2 shared challenges with lesson implementation around end-of-year testing. In addition, as indicated in the study findings, some teachers chose to implement lessons on the same days and times as their colleagues, whereas teachers at other schools chose to split up different days and times of the week to maximize the number of KIBO kits available to each classroom. These logistical differences made an impact on teachers' curriculum experiences and may have influenced their attitudes and beliefs about teaching coding and robotics. However, classrooms were observed at multiple points during the study, and the research team also checked in regularly with the district coordinator to understand how lessons were going for teachers. These alternate activities, alongside teachers' completion of lesson logs, provided adequate indication of teachers' curriculum implementation.

Validity

The rigor of qualitative research is measured by the reliability and validity of findings. In this study, two independent researchers reviewed and coded the full dataset and met multiple times to review the codes and refine the codebook. Inter-rater reliability was measured after the initial review of the full dataset, which exceeded the standard benchmark for adequate reliability. The strong correspondence between coders also adds to the validity of findings. Another way of ensuring validity of thematic analysis is to present the themes back to the original participants and ask whether they "see" themselves represented in the data. Due to time constraints, this method of checking validity was not performed and thus is a study limitation. However, preliminary findings and reports were shared and corroborated by the district coordinator, which provides some evidence of validity. Future work should aim to involve teachers in the analytic process, or at the very least, review and provide feedback on themes to ensure validity of findings.

Recommendations for Future Work

Other Grades

This study was replicated in first grade classrooms in the same set of elementary schools in the Norfolk Public School district, but with slightly modified study protocols. Teachers participated in one-on-one interviews, as well as focus groups with other members of their grade level team, at similar timepoints as this study. It would be an interesting follow-up study to examine these teachers' reported experiences and attitudes and see how they compare to this study's findings. In addition, the CAL-KIBO curriculum has been recently expanded and adapted for kindergarten and pre-kindergarten. Future work might explore how teachers' knowledge, attitudes, beliefs, and practices vary by grade level.

Replications and Extensions

Current work at the DevTech Research Group has focused on replicating CAL with a different coding platform, the ScratchJr programming application. Two different U.S. school districts are participating in this research and currently in the process of implementing the CAL-ScratchJr curriculum in kindergarten, first grade, and second grade classrooms. Teacher interviews and focus groups are being conducted by an external evaluation team. Further analyses of these data might benefit from using these findings as a starting point to investigate teachers' experiences and attitudes around participating in the CAL-ScratchJr program. This extended study not only encompasses multiple grade levels, but also incorporates a much larger sample size of teachers and students. Future work might aim to replicate the CAL-KIBO program on a similar scale, which would present the opportunity to conduct a comparative study to explore whether the type of coding interface might impact educators' experiences and attitudes.

Concluding Remarks

Researchers, educators, policymakers, and families increasingly value CS education and view learning to code as a key component of children's future success (Google & Gallup, 2020). This growing momentum around CS education has led to many questions about the preparedness of early elementary teachers to teach coding to their students, in addition to the availability of resources to support teachers in this endeavor. Whereas only a handful of coding tools and curriculum resources existed less than a decade ago, the CS education field has collectively made great strides in creating developmentally appropriate coding and robotics technologies, curriculum resources, and professional development opportunities for early elementary teachers and students. However, limited research has focused on teachers' coding experiences and attitudes beyond professional development workshops, after teachers have had a chance to formally integrate coding into their curriculum instruction. Findings from this dissertation indicate that there is much to understand about teachers' experiences and attitudes around coding and robotics education. "Computer science for all" will not become a reality until we understand teachers' experiences with teaching coding and address the ways in which teachers' beliefs and attitudes impact quality CS learning.

References

Ackermann, E. (2001). Piaget's Constructivism, Papert's Constructionism: What's the difference. *Future of Learning Group Publication, 5*(3), 438.

Agee, J. (2019). Developing qualitative research questions: a reflective process. *International Journal of Qualitative Studies in Education, 22*(4), 431-447, DOI: 10.1080/09518390902736512

Albo-Canals, J., Barco, A., Relkin, E., Hannon, D., Heerink, M., Heinemann, M., Leidl, K., & Bers, M. (2018). A Pilot Study of the KIBO Robot in Children with Severe ASD. *International Journal of Social Robotics, 10*(3), 371-383. DOI: 10.1007/s12369-018-0479-2

Alexander, C., Knezek, G., Christensen, R., Tyler-Wood, T. & Bull, G. (2014). The Impact of Project-Based Learning on Pre-Service Teachers' Technology Attitudes and Skills. *Journal of Computers in Mathematics and Science Teaching, 33*(3), 257-282.

Armstrong, K., Cusumano, D. L., Todd, M. & Cohen, R. (2008). Literacy Training for Early Childhood Providers: Changes in Knowledge, Beliefs, and Instructional Practices. *Journal of Early Childhood Teacher Education, 29*(4), 297-308. DOI: 10.1080/10901020802470085

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?. *Acm Inroads, 2*(1), 48-54.

Benitti, F. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers and Education, 58,* 978-988.

Bers, M. U. (2018). *Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom.* New York, NY: Routledge Press.

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

Bers, M. (2021). Coding, robotics and socio-emotional learning: developing a palette of virtues PIXEL-BIT. *Revista de Medios y Educación, 62,* 309-322.

Bers, M., Govind, M., & Relkin, E. (2021). Coding as Another Language: Computational Thinking, Robotics, and Literacy in First and Second Grade. In Ottenbreit-Leftwich, A., & Yadav, A. (2021). *Computational Thinking in PreK-5: Empirical Evidence for Integration and Future Directions.* ACM and the Robin Hood Learning + Technology Fund, New York, NY.

Bers, M. U., González-González, C., & Armas–Torres, M. B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers & Education, 138,* 130-145.

Bers, M., Ponte, I., Juelich, K., Viera, A., Schenker, J. (2002). Teachers as Designers: Integrating Robotics in Early Childhood Education. *Information Technology in Childhood Education.* 123-145.

Bers, M. U., Seddighin, S., & Sullivan, A. (2013). Ready for robotics: Bringing together the T and E of STEM in early childhood teacher education. *Journal of Technology and Teacher Education, 21*(3), 355-377.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology, 3*(2), 77–101. DOI: 10.1191/1478088706qp063oa

Bronfenbrenner, U. (2005). *Making human beings human: Bioecological perspectives on human development.* Sage Publications.

Brosterman, N. (1997). *Inventing kindergarten.* New York, NY: Harry N. Abrams Inc.

Bryan, L., & Atwater, M. (2002), Teacher beliefs and cultural models A challenge for science teacher preparation programs. *Science Education, 86,* 821-839.

Burke, Q., Roschelle, J., Bailey, C., Angevine, C., Weisgrau, J., & Mills, K. (2020). Examining Teacher Perspectives on Computational Thinking in K-12 Classrooms. Proceedings of *Research on Equity and Sustained Participation in Engineering, Computing, and Technology.* DOI: 10.1109/RESPECT49803.2020.9272483

Century, J., Ferris, K.A. & Zuo, H. (2020). Finding Time For Computer Science In the Elementary School Day: A Quasi-Experimental Study of a Transdisciplinary Problem-Based Learning Approach. *International Journal of STEM Education, 7*(20). DOI: 10.1186/s40594-020-00218-3

Çetin, M. & Demircan, H. O. (2020). Empowering technology and engineering for STEM education through programming robots: a systematic literature review. *Early Child Development and Care, 190*(9), 1323-1335, DOI: 10.1080/03004430.2018.1534844

Chalmers, C. (2018). Robotics and computational thinking in primary school. *International Journal of Child-Computer Interaction, 17,* 93-100. DOI: 10.1016/j.ijcci.2018.06.005

Chen, J., & Chang, C. (2006). Using computers in early childhood classrooms: Teachers' attitudes, skills, and practices. *Journal of Early Childhood Research, 4*(2), 169-188. DOI: 10.1177/1476718X06063535

Code.org (2021). More Data and Talking Points for Advocacy. https://code.org/promote/morestats

Code.org, CSTA, & ECEP Alliance (2021). 2021 State of computer science education: Accelerating action through advocacy. https://advocacy.code.org/stateofcs

Clements, D., & Sarama, J. (1997). Research on Logo. *Computers in the Schools, 14*(1), 9-46. DOI: 10.1300/J025v14n01_02

Creswell, J. W., Plano Clark, V. L., Gutmann, M., & Hanson, W. (2003). Advanced mixed methods research designs. In A. Tashakkori & C. Teddlie (Eds.), *Handbook of mixed methods in social and behavioral research* (pp. 209–240). Thousand Oaks, CA: Sage Publications, Inc.

Creswell, J. W. (2007). *Qualitative inquiry and research design: Choosing among five approaches (2nd ed.).* Sage Publications, Inc.

Darling-Hammond, L. (2000). Teacher Quality and Student Achievement. *Education Policy Analysis Archives, 8,* 1. DOI: 10.14507/epaa.v8n1.2000

Desimone, L. (2009). Improving Impact Studies of Teachers' Professional Development: Toward Better Conceptualizations and Measures. *Educational Researcher, 38*(3), 181-199. DOI: 10.3102/0013189X08331140

EdTech Evidence Exchange (2021). The EdTech Genome Project Report. https://edtechevidence.org/wp-content/uploads/2021/07/1.-FINAL-EdTechGenomeProject-FinalReport_July2021-2.pdf

Elkin, M., Sullivan, A., & Bers, M. U. (2016). Programming with the KIBO Robotics Kit in Preschool Classrooms. *Computers in the Schools, 33*(3), 169-186. DOI: 10.1080/07380569.2016.1216251

Ensign, T. I. (2017). Elementary Educators' Attitudes about the Utility of Educational Robotics and Their Ability and Intent to Use It with Students. Graduate Theses, Dissertations, and Problem Reports. 5546. https://researchrepository.wvu.edu/etd/5546

Ernest, P. (1989). The Knowledge, Beliefs and Attitudes of the Mathematics Teacher: a model. *Journal of Education for Teaching: International Research and Pedagogy, 15*(1), 13-33. DOI: 10.1080/0260747890150102

Ertmer, P. A., & Ottenbreit-Leftwich, A. T. (2010). Teacher technology change: how knowledge, beliefs, and culture intersect. *Journal of Research on Technology in Education, 42,* 255–284.

Fayer, S., Lacey, A. & Watson, A. (2017). STEM Occupations: Past, Present, And Future. https://www.bls.gov/spotlight/2017/science-technology-engineering-and-mathematics-stem-occupations-past-present-and-future/pdf/science-technology-engineering-and-mathematics-stem-occupations-past-present-and-future.pdf

Fenty, N. & McKendry, E. (2014). Examining Educators' Knowledge, Beliefs, and Practices About Using Technology With Young Children. *Journal of Early Childhood Teacher Education, 35*(2), DOI: 10.1080/10901027.2014.905808

Fluck, A., Webb, M., Cox, M., Angeli, C., Malyn-Smith, J., Voogt, J., & Zagami, J. (2016). Arguing for Computer Science in the School Curriculum. *Educational Technology & Society, 19*(3), 38–46.

Giannakos, M., Doukakis, S., Pappas, I., Adamopoulos, N., & Giannopoulou, P. (2015). Investigating teachers' confidence on technological pedagogical and content knowledge: an initial validation of TPACK scales in K-12 computing education context. *Journal of Computers in Education, 2*(1), 43-59. DOI: 10.1007/s40692-014-0024-8

Google & Gallup (2020). Current Perspectives and Continuing Challenges in Computer Science Education in U.S. K-12 Schools. https://services.google.com/fh/files/misc/computer-science-education-in-us-k12schools-2020-report.pdf

Govind, M., Hassenfeld, Z. & de Ruiter, L. (2021). Rhyme and Reason: The Connections Among Coding, Computational Thinking, and Literacy. In M. U. Bers (Ed.), *Teaching Computational Thinking and Coding to Young Children.* IGI Global.

Grauer, K. (1998). Beliefs of Preservice Teachers toward Art Education. *Studies in Art Education, 39*(4), 350-370. DOI: 10.1080/00393541.1998.11650037

Gresham, G. & Burleigh, C. (2019). Exploring early childhood preservice teachers' mathematics anxiety and mathematics efficacy beliefs. *Teaching Education, 30*(2), 217-241. DOI: 10.1080/10476210.2018.1466875

Grover, S. & Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher, 42*(1), 38-43. DOI: 10.3102/0013189X12463051

Hammond, L. (2015). Early childhood educators' perceived and actual metalinguistic knowledge, beliefs and enacted practice about teaching early reading. *Australian Journal of Learning Difficulties, 20*(2), 113-128. DOI: 10.1080/19404158.2015.1023208

Hardy, J. V. (1999). Teacher Attitudes Toward and Knowledge of Computer Technology. *Computers in the Schools, 14,* 119-136. DOI: 10.1300/J025v14n03_11

Hassenfeld, Z. R., Govind, M., de Ruiter, L. E., & Bers, M. U. (2020). If You Can Program, You Can Write: Learning Introductory Programming Across Literacy Levels. *Journal of Information Technology Education: Research, 19,* 65-85. DOI: 10.28945/4509

Invernizzi, M., Meier, J., & Juel, C. (2015). Phonological Awareness Literacy Screening 1-3 Technical Reference. https://pals.virginia.edu/public/pdfs/rd/tech/1-3_Tech_Ref_2015.pdf

Israel, M., Pearson, J., Tapia, T., Wherfel, Q., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers and Education, 82,* 263-279. DOI: 10.1016/j.compedu.2014.11.022

Jaipal-Jamani, K. & Angeli, C. (2017). Effect of Robotics on Elementary Preservice Teachers' Self-Efficacy, Science Learning, and Computational Thinking. *Journal of Science Education and Technology, 26,* 175-192. DOI: 10.1007/s10956-016-9663-z

Judson, E. (2006). How Teachers Integrate Technology and Their Beliefs About Learning: Is There a Connection? *Journal of Information Technology for Teacher Education, 14*(3).

Jung, S. E. & Won, E. (2018). Systematic Review of Research Trends in Robotics Education for Young Children. *Sustainability, 10,* 905. DOI: 10.3390/su10040905

K–12 Computer Science Framework (2016). Retrieved from http://www.k12cs.org

Kafai, Y. B., & Burke, Q. (2014). *Connected code: Why children need to learn programming.* MIT Press.

Kaleli, Y. S. (2020). Investigation of the relationship between pre-service music teachers' attitudes towards teaching profession and their self-efficacy beliefs. *International Journal of Research in Education and Science (IJRES), 6*(4), 580-587.

Khanlari, A. (2016). Teachers' perceptions of the benefits and the challenges of integrating educational robots into primary/elementary curricula. *European Journal of Engineering Education, 41*(3), 320-330. DOI: 10.1080/03043797.2015.1056106

Kim, C., Kim, D., Yuan, J., Hill, R. B., Doshi, P., & Thai, C. N. (2015). Robotics to promote elementary education pre-service teachers' STEM engagement, learning, and teaching. *Computers and Education*, 14-31. DOI: 10.1016/j.compedu.2015.08.005

Kong, S. T. (2019). A Case Study on Elementary Teachers' Experiences Teaching Computer Science (Thesis, Concordia University, St. Paul). https://digitalcommons.csp.edu/cup_commons_grad_edd/316

Little, J. W. (1987). Teachers as colleagues. In V. Richardson-Koehler (Ed.), *Educators' handbook: A research perspective* (pp. 491–518). New York: Longman.

Lye, S. Y. & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior, 41,* 51-61.

Maier, M. F., Greenfield, D. B., & Bulotsky-Shearer, R. J. (2013). Development and validation of a preschool teachers' attitudes and beliefs toward science teaching questionnaire. *Early Childhood Research Quarterly, 28*(2), 366-378. DOI: 10.1016/j.ecresq.2012.09.003

Madkins, T. C., Howard, N. R., & Freed, N. (2020). Engaging Equity Pedagogies in Computer Science Learning Environments. *Journal of Computer Science Integration, 3*(2), 1-27. DOI: 10.26716/ jcsi.2020.03.2.1

Mason, S. L., & Rich, P. J. (2019). Preparing elementary school teachers to teach computing, coding, and computational thinking. *Contemporary Issues in Technology and Teacher Education, 19*(4), 790-824.

McAlister, A. M., Lee, D. M., Ehlert, K. M., Kajfez, R. L., Faber, C. J., & Kennedy, M. S. (2017). *Qualitative Coding: An Approach to Assess Inter-Rater Reliability* [Paper presentation]. American Society for Engineering Education. DOI: 10.18260/1-2--28777

Meacham, S., & Atwood-Blaine, D. (2018). Early Childhood Robotics: A Lego robotics club inspired by Reggio Emilia supports children's authentic learning. *Science and Children, 56*(3), 57-63.

Menekse, M. (2015). Computer science teacher professional development in the United States: a review of studies published between 2004 and 2014. *Computer Science Education, 25*(4), 325-350. DOI: 10.1080/08993408.2015.1111645

Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook* (Second Edition). Sage Publications.

Mishra, P., & Koehler, M. J. (2006). Technological Pedagogical Content Knowledge: A new framework for teacher knowledge. *Teachers College Record, 108*(6), 1017–1054.

Ottley, J. R., Piasta, S. B., Mauck, S. A., O'Connell, A., Weber-Mayrer, M., & Justice, L. M. (2015). The nature and extent of change in early childhood educators' language and literacy knowledge and beliefs. *Teaching and Teacher Education, 52,* 47-55.

Papadakis, S., Vaiopoulou, J., Sifaki, E., Stamovlasis, D., & Kalogiannakis, M. (2021). Attitudes towards the Use of Educational Robotics: Exploring Pre-Service and In-Service Early Childhood Teacher Profiles. *Education Sciences, 11,* 204. DOI: 10.3390/educsci11050204

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas.* New York: Basic Books.

Patton, M. Q. (2015). *Qualitative Evaluation and Research Methods.* Thousand Oaks, CA: Sage.

Phillips, R. S. & Brooks, B. (2017). The Hour of Code: Impact on Attitudes Towards and Self-Efficacy with Computer Science. https://code.org/files/HourOfCodeImpactStudy_Jan2017.pdf

Podolsky, A., Darling-Hammond, L., Doss, C., & Reardon, S. (2019). *California's positive outliers: Districts beating the odds.* Palo Alto, CA: Learning Policy Institute.

Pugnali, A., Sullivan, A., & Bers, M. U. (2017). The Impact of User Interface on Young Children's Computational Thinking. *Journal of Information Technology Education: Innovations in Practice, 16,* 172-193.

Relkin, E. & Bers, M. U. (2020). Exploring the Relationship Between Coding, Computational Thinking and Problem Solving in Early Elementary School Students. *Annual Meeting of the American Educational Research Association (AERA)*, San Francisco, CA.

Relkin, E., de Ruiter., L., & Bers, M. U. (In Press, 2020). TechCheck: Development and Validation of an Unplugged Assessment of Computational Thinking in Early Childhood Education. *Journal of Science Education and Technology.* DOI: 10.1007/s10956-020-09831-x

Rice, J. K. (2003). *Teacher Quality: Understanding the Effectiveness of Teacher Attributes.* Economic Policy Institute.

Rich, P. J., Jones, B. L., Belikov, O., Yoshikawa, E., & Perkins, M. (2017). Computing and Engineering in Elementary School: The Effect of Year-long Training on Elementary Teacher Self-efficacy and Beliefs About Teaching Computing and Engineering. *International Journal of Computer Science Education in Schools, 1*(1). DOI: 10.21585/ijcses.v1i1.6

Rivkin, S. G., Hanushek, E. A., & Kain, J. F. (2005). Teachers, Schools, and Academic Achievement. *Econometrica, 73*(2), 417-458.

Schachter, R. E., Spear, C. F., Piasta, S. B., Justice, L. M., & Logan, J. (2016). Early childhood educators' knowledge, beliefs, education, experiences, and children's language- and literacy-learning opportunities: What is the connection? *Early Childhood Research Quarterly, 36,* 281-294. DOI: 10.1016/j.ecresq.2016.01.008

Seiter, L., & Foreman, B. (2013). Modeling the learning progressions of computational thinking of primary grade students. *Proceedings of the ACM Conference on International Computing Education Research.* DOI: 10.1145/2493394.2493403

Shulman, L. S. (1986). Knowledge and teaching: Foundations of the new reform. *Harvard Educational Review, 57*(1), 1–22.

Stanton, J., Goldsmith, L., Adrion, W. R., Dunton, S., Hendrickson, K. A., Peterfreund, A., ... Zinth, J. D. (2017). State of the states landscape report: State- level policies supporting equitable K–12 computer science education. https://www.edc.org/state-states- landscape-report-state-level-policies-supporting-equitable-k-12-computer- science

Strawhacker, A. L., & Bers, M. U. (2015). "I want my robot to look for food": Comparing children's programming comprehension using tangible, graphical, and hybrid user interfaces. *International Journal of Technology and Design Education, 25*(3), 293-319. DOI: 10.1007/s10798-014-9287-7

Strawhacker, A., & Bers, M. U. (2019). What They Learn When They Learn Coding: Investigating cognitive domains and computer programming knowledge in young children. *Educational Technology Research and Development, 67*(3), 541-575. DOI: 10.1007/s11423-018-9622-x

Strawhacker, A. L., Lee, M. S. C., & Bers, M. U. (2017). Teaching tools, teachers' rules: exploring the impact of teaching styles on young children's programming knowledge in ScratchJr. *International Journal of Technology and Design Education.* DOI: 10.1007/s10798-017-9400-9

Strawhacker, A., Sullivan, A., & Bers, M.U. (2013). TUI, GUI, HUI: Is a bimodal interface truly worth the sum of its parts?: *Proceedings of the 12th International Conference on Interaction Design and Children (IDC '13)* (pp. 309-312). New York, NY: ACM.

Sullivan, A. A. (2019). *Breaking the STEM stereotype: Reaching girls in early childhood.* Lanham, MD: Rowman & Littlefield Publishers.

Sullivan, A., & Bers, M. U. (2017). Dancing robots: Integrating art, music, and robotics in Singapore's early childhood centers. *International Journal of Technology and Design Education.* DOI: 10.1007/s10798-017-9397-0

Sullivan, A., Bers, M. U., Mihm, C. (2017). Imagining, Playing, & Coding with KIBO: Using KIBO Robotics to Foster Computational Thinking in Young Children. Proceedings of the *International Conference on Computational Thinking Education.* Wanchai, Hong Kong.

Sullivan, A., Elkin, M., & Bers, M. U. (2015). KIBO Robot Demo: Engaging young children in programming and engineering. *Proceedings of the 14th International Conference on Interaction Design and Children (IDC '15)*, Medford, MA, June 21-25. New York, NY: ACM.

Taber, K. S. (2018). The Use of Cronbach's Alpha When Developing and Reporting Research Instruments in Science Education. Research in Science Education, 48(1), 1-24. DOI: 10.1007/s11165-016-9602-2

Taylor, V. S. (2021). Investigating Pre-Service Teachers' Perceptions of the Virginia Computer Science Standards of Learning: A Qualitative Multiple Case Study. Doctor of Philosophy (PhD) Dissertation, Teaching and Learning, Old Dominion University. DOI: 10.25777/jnm7-f468

Toh, L. P. E., Causo, A., Tzuo, P. W., Chen, I. M., & Yeo, S. H. (2016). A Review on the Use of Robots in Education and Young Children. *Educational Technology & Society, 19*(2), 148–163.

Tucker, A. (Ed.), Deek, F., Jones, J., McCowan, D., Stephenson, C., & Verno, A. (2003, October). *A model curriculum for K–12 computer science: Final report of the ACM K–12 task force curriculum committee.* New York, NY: Association for Computing Machinery.

U.S. Department of Education, Office of Educational Technology (2017). Reimagining the role of technology in education: 2017 National Education Technology Plan update. https://tech. ed.gov/teacherprep

Valenzuela, J. (2019). Attitudes Towards Teaching Computational Thinking and Computer Science: Insights from Educator Interviews and Focus Groups. *Journal of Computer Science Integration, 2*(2), 1-17. DOI: 10.26716/jcsi.2019.02.2.2

Van Driel, J. H., Beijaard, D., & Verloop, N. (2001). Professional development and reform in science education: The role of teachers' practical knowledge. *Journal of Research in Science Teaching, 38,* 137-158.

Virginia Department of Education (2021). Computer Science Instructional Resources. https://www.doe.virginia.gov/instruction/computer-science/index.shtml

Wilkins, J. L. M. (2008). The Relationship among Elementary Teachers' Content Knowledge, Attitudes, Beliefs, and Practices. *Journal of Mathematics Teacher Education, 11,* 139-164.

Wilson, S. M., Shulman, L. S., & Richert, A. (1987). 150 different ways of knowing: Representations of knowledge in teaching. In J. Calderhead (Ed.), *Exploring teachers' thinking* (pp. 104-124). Sussex, England: Holt, Rinehart & Winston.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35. https://www.cs.cmu.edu/*15110-s13/Wing06-ct.pdf

Wohl, B., Porter, B., & Clinch, S. (2015). Teaching Computer Science to 5-7 year-olds: An initial study with Scratch, Cubelets and unplugged computing. *Workshop in Primary and Secondary Computing Education.* DOI: 10.1145/2818314.2818340

Yadav, A., Gretter, S., Hambrusch, S. & Sands, P. (2016). Expanding computer science education in schools: understanding teacher experiences and challenges. *Computer Science Education, 26*(4), 235-254. DOI: 10.1080/08993408.2016.1257418

Yadav, A., Krist, C., Good, J. & Caeli, E. N. (2018). Computational thinking in elementary classrooms: measuring teacher understanding of computational ideas for teaching science. *Computer Science Education, 28*(4), 371-400. DOI: 10.1080/08993408.2018.1560550

Yu, J., & Roque, R. (2018). A survey of computational kits for young children. *Proceedings of the 17th ACM Conference on Interaction Design and Children - IDC '18* (pp. 289–299). Trondheim, Norway: ACM Press. DOI: 10.1145/3202185.3202738

Appendix A. Semi-Structured Interview Protocol

Interviewer Notes: Ask these questions as a starting point for conversation along with any relevant follow-up questions and prompts. This is a semi-structured interview, meaning that these questions should be used as a guide, but other relevant topics can be addressed. Participants may skip any questions they do not feel comfortable answering. Interviewers may skip/add/adapt any questions as needed.

[Instructions for recording phone conversations omitted]

Pre-Training Interview Questions

1. Tell me a little bit about your role at the school. How many years have you been teaching overall? How many years have you been teaching at this school? What has been your experience thus far with coding?
2. So far, how do coding and/or robotics fit into the rest of your classroom curriculum? What has been challenging? What has been easy?
   Probe: In which part of the day does this fit in? How many hours/days per week? Do all students engage in these activities?
3. What is most important to you during a professional development training? Least important?
4. What are you looking forward to about the KIBO robotics curriculum? What do you anticipate being the challenges?
5. What are your priorities in literacy instruction? What is the range of literacy abilities/levels in your class and how do you serve the diversity of levels/needs? How do you use the PALS assessment in your classroom?
   Probe: How many students get additional/remedial literacy instruction?
6. Is there anything else you'd like to share?

Post-Training Interview Questions

1. What is your reaction so far to KIBO?
2. What are you looking forward to most when you begin the coding curriculum with your students?
3. What are you most nervous about before you begin the coding curriculum with your students?
4. What kind of support would be most helpful as you prepare to start teaching?
5. What kind of support would be least helpful as you prepare to start teaching?
6. Is there anything else you'd like to share?

Pre-Curriculum Interview Questions

1. What is your reaction so far to KIBO?
2. What are you looking forward to most when you begin the coding curriculum with your students?
3. What are you most nervous about before you begin the coding curriculum with your students?
4. What kind of support would be most helpful as you prepare to start teaching?
5. What kind of support would be least helpful as you prepare to start teaching?
6. Is there anything else you'd like to share?

Mid-Curriculum Interview Questions
1. Tell me a little bit about where you are in the robotics curriculum.
2. What is your reaction so far to KIBO?
3. What has been challenging? What has been easy?
4. So far, how does these activities fit into the rest of your classroom curriculum?
5. What do you anticipate will be the challenges you will face as you progress into the later lessons of the curriculum?
6. What pedagogical tools/strategies have you been using so far?
7. Is there anything else you'd like to share?

Post-Curriculum Interview Questions
1. Now that you have completed the full coding curriculum, what would you say were the highlights of your experience?
2. What were some of the major challenges you faced in implementing this curriculum?
3. In what ways did you feel supported, or not so supported, throughout the curriculum?
4. What pedagogical tools/strategies did you use that were most effective? Least effective?
5. Would you participate in something like this again? If so, how would you want your experience to be similar or different?
6. Is there anything else you'd like to share?

Appendix B. Semi-Structured Interview Codebook

Knowledge

Category	Sub-Category	Code	Example Quote
Coding and Robotics	Gains	KIBO knowledge	I learned how to interact with a KIBO robot
		Screen-free coding	I learned that coding doesn't just involve sitting in front of a computer and typing things
	Challenges	Difficult KIBO concepts	Yeah, we had a really hard time with that, and so I couldn't figure out for a while. Are we doing the conditional statements wrong?
		Lack of knowledge	I just don't know robotics. I have no idea
Teaching	Classroom management	-	Unless you have that strong classroom management of that routine structure, it'll be hard to get the students back on track

Note. This codebook only includes teacher knowledge. Teachers' reflections on student knowledge gains and challenges were omitted for this analysis.

Attitudes

Category	Sub-Category	Code	Example Quote
KIBO	Positive	Excitement around specific KIBO concepts	the highlight for them, especially was getting it to sing and the lights to come on and off
		Hands-on, engaging, user-friendly tool	They like how engaging it is, how hands-on it is, and that's what I like about it too
	Negative	Concern about moving robots in/out of classroom	first, we have to get the carts, bring them in and out
		Students' mixed reactions to KIBO	at first, they were really excited, but then as they got used to KIBO, the excitement eventually waned off a little bit
Teaching Coding and Robotics	Positive	Personal excitement	I'm really excited to get back to and show my second graders how to do it
		Open to learning	I'm open to learning and trying it in here, absolutely
		Student excitement	The kids were super engaged, they're very excited about using KIBO

		Receptive to technology usage	I use the computers and I love them, and I use the Smart Board and all that
		Feeling prepared and confident	I feel pretty confident in everything, just because we've had a lot of support
	Negative	Nervous about lack of experience	I'm a little nervous just about something new, but I get nervous about doing new things anyway
		Nervous about answering student questions	I like to be able to explain it when the kids have questions
		Nervous about students working together	I was really nervous about them being able to collaborate
		Resistant to technology usage	I was very reluctant to come candid because I'm just not so technically savvy
		Concern about curriculum alignment	I'm sure there is the possibility that it could be more cohesively kind of aligned, but it would probably be more time consuming
Literacy Integration	Positive	Receptive to literacy integration	I like how it relates to sequencing in reading 'cause that's something we're constantly talking about which is, it has to be in order or it's not gonna makes sense
		Implement in ELA block	It took up our writing time for the day, and then some of our reading curriculum time.
	Negative	Resistant to literacy integration	I don't think I would do my shared reading again, just because we had to cover so many standards of learning
		Implement in non-ELA block	I'm taking that [computer] time and then taking a little bit of Social Studies and Science time
Non-literacy Integration	Positive	Receptive to non-literacy integration	We talked about how it relates to life cycles of science, 'cause we've been going through metamorphosis for butterflies and things like that

Beliefs

Category	Sub-Category	Code	Example Quote

Coding	Overall benefits	Student creativity	I liked the creative portion of the writing whenever they got to be creative, they really enjoyed that
		Learning in other areas	I could see this being beneficial, I could see this really changing how we look at science, how we look at even reading
		Student independence	they were like, "Yeah, I'm in charge and I can figure this out, I'm the programmer, I'm the coder"
		New ways of thinking	I see a need for it. And I do think it's the wave of the future. And I do think the robotics will be a part of it and in this higher-level thinking
		Problem solving skills	I feel like the students have become better problem solvers in all aspects of the day
		Important 21st century skill	I think it's definitely a necessity for the 21st century learner and the children will benefit
	Benefits for young children	Important to introduce early	I think it's important and I feel like they need to start young
		Young children will pick up coding easily	the younger you are able to access new information to students, the quicker they learn. Second graders have been exposed to technology for a long period of time already in their lives, and so I think they'll pick this up very, very quickly
Coding and Inclusion	Positive	Coding should be part of regular curriculum	I wish it was almost like an hour, or our engineering hour, or part of our curriculum
		Coding for students of all backgrounds and ability levels	I think that this sort of, kind of evens the playing field and allows students with different learning techniques to be able to learn and work well
		Some inclusion students excel in coding	I was very shocked because he's normally not as verbal and knowledgeable about a lot of things. And I did not know that he was very, very much so
		Disrupt gender stereotypes around coding	I feel there's always a stigma like, with STEM and computer science is usually more male-dominated, but the girls in my class took charge, and I was like, "Yeah, girl power"
	Negative	Some inclusion students	She may need help with the blocks, reminding her that the different things on the blocks show you what it stands for

		struggle with coding	
Education	Learning as ongoing, iterative process	Lifelong learner mindset	we don't have to stop learning at any point in our lives, that we can continue learning for the rest of our lives
		Not knowing answer right away	That was like the one block that I was just not feeling it, not getting it quite right, but we worked with it… We weren't gonna quit because it didn't make sense to us
		Making mistakes as key to learning	they were a little more frustrated…when they had an issue with KIBO or challenge. But I feel like as we progress through the lessons that they've learned that it's just gonna happen
	Fun	Important to have fun while learning	I want them to have fun and learn at the same time without realizing that, "Hey this is science"

Support

Category	Sub-Category	Code	Example Quote
Human Supports	-	School team	The four of us on the second grade would talk 'cause I was like, "Guys, I don't get it." And they're like, "I don't get this one."
	-	Researcher-coach	I like the fact that you all visited the schools and the classrooms and provided us with the training
	-	ITRT	If we needed support from our ITRT person, she'd always come in, and every week when she was in the building and check in and just ask me, "How are you doing? When are you gonna do your lesson?"
	-	District coordinator	She was quick to give us any type of materials, and she would visit frequently so that we could access any kind of new information that we needed
	-	Children	I took my higher-level learners, and I taught them first… then they turned around and just taught the other kids
Material Supports	Visuals	Anchor charts	I love to use anchor charts, to create the anchor charts with the kids and put them on display
		Lesson slide decks	I created a smart board slide that basically had a header for each section, so it gave them something to look at, but it also gave me a mental cue
		Lesson videos	That KIBO blast thing was good, because before we had to teach the lesson, we were able to see a video
		Additional videos	I went to YouTube to do some research for it
		Cleanup visual	Something that would just help them have a clear idea of where it goes. Some sort of label
	Curriculum (format and structure)	Positive	I liked the progression of it. It seems like it was very well thought out and it just flows
		Negative	it almost seemed like the verbiage was too high for the kids… I broke it down to everyday language so that they would be familiar with it and comfortable with it
	Curriculum (lesson activities)	Positive	The songs are great, by the way...I mean, they knew the design process before I did
		Negative	it'd probably be more beneficial if they had something that they could have in their hands to read

	-	Arts and crafts supplies	One of the things that I would have suggested for the future is to include an arts and crafts kit

Behaviors

Category	Sub-Category	Code	Example Quote
Pedagogical Strategies	Collaborative learning	Share programs with peers	I made sure we all came back together and we're able to share our work in the tech circle
		Talk through programming challenges	we would always brainstorm, and I would model, but also giving students a chance to talk to each other, just to talk out like… "Why doesn't this code work?"
		Work in small groups	I want the kids to be working in pairs
	Applied learning	Activate prior knowledge	in other robots that are coding robots that they played with, they had the clap feature, they could relate to it
		Make real-world connections	There was really no formal beginning, no formal ending to either one. And we discussed ways how that occurs in real life
	Scaffolded learning	Learn through debugging programs	They learned how to properly do it so that there were no kinks. And if they did have a kink, they would go back and figure it out
		Plan before programming	we would've benefited from doing more of the planning with your partner and recording what the plan was, rather than just jumping down into it
Lesson Implementation	Strengths	KIBO part of weekly routine	I consistently taught it Mondays and Wednesdays 9:10 to 10:10. Still thought it was helpful having that designated spot
		Break up lesson into chunks	I would break it up into chunks, so the writing piece would be done during our writing time and then they would have time to explore
		Teachers implement on different days	“Okay, why don't we have somebody do Monday Wednesday and somebody else do Tuesday Thursday?" And then it was just a set schedule
		Prepare for lessons	I would read it, I would preview it the week before, but I also, I'm very forgetful, so just as looking over what I

			need in the morning while as I'm prepping things made it a lot easier
	Challenges	Teachers implement on same days	if we hadn't been doing it all at the same time, then we could have used all the KIBOs
		Open classroom layout was disruptive	we have four classes in one room… It was distracting to the other teachers as well. But that's due to how our school is structured
	Challenges (time)	Concern about time constraints	I'm just trying to fit it all in and using my time the best I can to fit all my pieces together. And that's been my struggle
		Prep time takes longer than expected	It took longer than I expected for the preparing
		Lessons take longer than expected	Oftentimes, the lesson ends up taking longer
		Cleanup takes longer than expected	I cut KIBO down from an hour to I think it was like maybe 40 minutes…Then we got the math done
		Shorten lesson to focus on other content	it's just what takes up a little more time trying to make sure everything gets in there the way it should
		Need more time before implementation	I feel like we're jumping into this a little bit too fast
KIBO Organization	Strengths	Central KIBO storage space	we store them down in the library
	Challenges	Put away kits properly	one of the challenges that came up was the mixing of the KIBO kits
		Share kits with other classrooms	we go pick 'em up, bring 'em back, gotta do the lesson, and then we walk 'em to the other teacher's classroom

Appendix C. Details of Full Study Measures

Measure	Participant	Description	Administration and Scoring
Surveys	Teachers	Measure educators' self-reported (1) general coding knowledge; (2) pedagogical content knowledge surrounding how to teach coding; (3) general KIBO robotics knowledge; (4) knowledge of specific KIBO sensors and modules; and (5) attitudes and self-efficacy surrounding the implementation of the CAL-KIBO curriculum.	• Administered at four timepoints: pre-training, post-training, pre-curriculum, and post-curriculum • 15-20 minutes to complete
Interviews	Teachers	Semi-structured questions related to teachers' experiences and attitudes surrounding KIBO and the CAL-KIBO curriculum.	• Administered at five timepoints: pre-training, post-training, pre-curriculum, mid-curriculum, and post-curriculum • 15-20 minutes
Classroom Observations	Teachers	Checklist that examines how the teacher facilitates students' positive behaviors when using technology. Behaviors consist of the "6 C's of positive technological development (PTD)": communication, collaboration, community building, content creation, creativity, and choice of conduct.	• Observational checklist • 6 sections, each representing a PTD framework behavior • Ratings on a 5-point Likert scale
Lesson Logs	Teachers	Measure implementation status of select CS curriculum modules. An example of a question is "Describe some of the challenging moments (if any) during this lesson".	• Teachers complete lesson logs at the end of each lesson • 5-10 minutes to complete
Tufts Assessment of Computational Thinking in	Teachers & Students	Classifies Computational Thinking abilities into seven domains and four proficiency levels. TACTIC is based upon	• 28-question computerized multiple-choice format • Designed for children ages 5-8

Children (TACTIC)		Bers' (2018) theoretical framework of seven powerful ideas of developmentally appropriate computer science. These seven ideas are Algorithms, Modularity, Hardware/Software, Control Structures, Debugging, Representation, and Design Process.	• 30-40 minutes to administer (with intermission) • Score range 0-28 • Yielding a 4-level composite rating of CT proficiency • Modified version for first grade has 21 items with score range 0-21, yielding 3-level composite rating of CT proficiency
Tech Check	Students	Formative assessment that uses "unplugged" (non-coding) tasks as a means of measuring problem solving abilities that may be enhanced by the acquisition of computational thinking skills.	• 15-question computerized multiple-choice format • Designed for children ages 5-8 • 10-20 minutes to administer • Score ranges from 0-15 • 2 alternate forms available
Design Journals	Students	Student workbook with planning, writing, and drawing activities corresponding to lesson activities.	• One per student • Collected at the end of the curriculum
KIBO Mastery Challenges (KMC)	Students	Formative assessment that measures children's KIBO knowledge as introduced in the classroom using the CAL-KIBO curriculum.	• 24-question multiple-choice format, paper-and-pencil • 4 sets of questions (A, B, C, D), each 6 items • Collected at the end of the curriculum

Where the Wild Things Are

A KIBO Robotics Coding Curriculum Integrated with Foundational Literacy Topics

Using the KIBO robotics kit and Coding as Literacy (CAL) approach developed by

DevTech Research Group
Eliot-Pearson Dept. of Child Study and Human Development
Tufts University

Table of Contents

INTRODUCTION

LESSONS

APPENDICES & REFERENCES

Introduction

CODING AS LITERACY (CAL) APPROACH

This curriculum introduces powerful ideas from computer science, specifically programming with KIBO robotics, to 2nd grade children in a structured, developmentally appropriate way. The **Coding as Literacy (CAL)** approach, developed by Prof. Marina Umaschi Bers and members of her DevTech Research Group at Tufts University, puts computer science ideas into direct conversation with powerful ideas from literacy. The starting assumption of the CAL curriculum is that both computer science and literacy can enhance one another. Instruction in both can be leveraged in service of the other. Both can support learners in developing new ways of thinking about themselves and the world.

Thinking involves the ability to make sense of, interpret, represent, model, predict, and invent our experiences in the world. Thus, as educators, we must give children one of the most powerful tools for thinking: language. The term **language** refers here to a system of communication, natural or artificial, composed of a formal limited system of signs, governed by syntactic and grammatical combinatory rules, that serves to communicate meaning by encoding and decoding information. Today, we have the opportunity to not only teach children how to think by using natural languages, such as English, but also by learning artificial languages—programming languages such as the one used by KIBO robots.

The achievement of literacy in a natural language involves a progression of skills beginning with the ability to understand spoken words, followed by the capacity to code and decode written words, and culminating in the deep understanding, interpretation, and production of text. The ultimate goal of literacy is not only for children to master the syntax and grammar, the orthography and morphology, but also the semantics and pragmatics, the meanings and uses of words, sentences and genres. A literate person knows that reading and writing are tools for meaning making and, ultimately, tools of power because they support new ways of thinking.

The CAL approach proposes that programming, as a literacy of the 21st century, engages new ways of thinking and new ways of communicating and expressing ideas, as well as new ways of problem solving and working with others. CAL understands the process of coding as a semiotic act, a meaning making activity that engages children in both developing computational thinking, as well as promoting personal expression, communication, and interpretation. This understanding shapes this curriculum and our strategies for teaching coding.

The curriculum is organized around **powerful ideas** from both computer science and literacy. The term powerful idea refers to a central concept or skills within a discipline that is simultaneously personally useful, inherently interconnected with other disciplines, and has roots in intuitive knowledge that a child has internalized over a long period of time. The **powerful ideas from computer science** addressed in this curriculum include: algorithms, design process, representation, debugging, control structures, modularity, and hardware/software. The **powerful ideas from literacy** that will be placed in conversation with these powerful ideas from computer science are: the writing process, recalling, summarizing and sequencing, using illustrative and descriptive language, recognizing literary devices such as repetition and foreshadowing, and using reading strategies such as predicting, summarizing, and evaluating.

The CAL approach allows students to make connections between coding and literacy and use the two platforms to express their thoughts and ideas. These powerful ideas of literacy and computer science are explored in the context of a curriculum that draws on the well-known children's book *Where the Wild Things Are* by Maurice Sendak, which is about a young boy named Max who makes mischief at home and then sails to the land where the wild things are.

Each lesson contains a variety of activities to introduce children to programming and literacy skills and concepts. Lessons are aligned to academic frameworks of Common Core, as well as Virginia Public Schools, as in 2017, Virginia became the first state in the US to formally mandate K-12 computer science standards. Teachers are encouraged to use this curriculum as a guiding resource and to adapt lessons and activities to their needs of their students. Activities in this curriculum include:

- Warm up games to playfully introduce or reinforce concepts
- Design challenges to introduce the powerful ideas from computer science
- Writing activities to introduce the powerful ideas from literacy
- Work individually or in pairs on designing and creating projects
- Technology circles to share and reflect on activities
- Free-explorations to allow students to tinker and expand their skills

The culmination of the unit is an open-ended project to share with family and friends. Just as young children can read age-appropriate books, computer programming can be made accessible by providing young children with appropriate tools such as KIBO.

PACING

This 12-hour curriculum unit is designed to take place over the course of a few months with one or two sessions per week (i.e. 1-2 hours each week for 2-3 consecutive months). This curriculum provides suggested time allotments, but they should be adapted to suit the needs of each classroom.

To supplement the structured challenges, free-exploration is allotted throughout the curriculum. These open-ended sessions are vital for children to fully understand the complex ideas behind their robotic creations and programs. The free-exploration sessions also serve as a time for teachers to observe students' progress and understandings. These sessions are as important for learning as the lessons themselves! In planning and adjusting the timeframe of this curriculum, free-exploration sessions should not be left by the wayside. Free-exploration provides opportunities for playing with materials and ideas. This will help build a solid foundation.

Table 1: Pacing Guide

Lesson	Activities
Lesson 1: Foundations	• What is an Engineer? (20 min) • Engineers and Writers (10 min) • Think Like an Engineer (10 min) • How to Build a Robot (20 min)
Lesson 2: Technological Tools - Robots	• Robot Corners (15 min) • Characteristics of Robots (10 min) • Tools of Communication (10 min) • Human Language vs. Code Language (10 min) • KIBO Says (15 min)
Lesson 3: Sequencing	• Where the Wild Things Are (20 min) • Order Matters (15 min) • Program the Teacher with KIBO Blocks (10 min) • Meet the KIBO Robot (15 min)

Lesson	Activities
Lesson 4: Programming	• Dance the Hokey-Pokey (5 min) • Program the Hokey-Pokey (20 min) • Hokey-Pokey Reflection (10 min) • Share Creations (10 min) • Solve-It Assessment A (15 min)
Lesson 5: Debugging	• Tell a Story (15 min) • Why is KIBO Confused? (15 min) • Free Play (20 min) • Debugging Reflection (10 min)
Lesson 6: Cause and Effect - Level 1	• What did Max Sense (15 min) • KIBO Sound Sensor (5 min) • Shape Shifting (15 min) • KIBO Sound Recorder (5 min) • Free Play (15 min) • Solve-It Assessment B (10 min)
Lesson 7: Cause and Effect - Level 2	• Sing “If You’re Wild and You Know It” (5 min) • Program "If You're Wild and You Know It" (30 min) • Project Reflection (10 min) • Share Creations (15 min)
Lesson 8: Repeat Loops - Level 1	• Repetition in Stories and Songs (15 min) • Toothbrush Exercise (15 min) • KIBO Repeat with Numbers (20 min) • Solve-It Assessment C (15 min)
Lesson 9: Repeat Loops - Level 2	• My Five Senses (20 min) • KIBO Repeat with Sensors (15 min) • Free Play with Repeats (25 min)
Lesson 10: If Statements	• Writing an Alternative Story (20 min) • KIBO If Statements (20 min) • Free Play with Conditionals (20 min)
Lesson 11: Final Project - Writing the Wild Rumpus Composition	• Wild Rumpus Composition (30 min) • Writing vs. Coding (5 min) • Peer Feedback (10 min) • Collaboration Web (5 min) • Begin Coding the Wild Rumpus (20 min)
Lesson 12: Final Project - Coding the Wild Rumpus	• Coding the Wild Rumpus (20 min) • Share Creations and Deliver Cards (15 min) • Wild Rumpus Reflection (10 min) • Solve-It Assessment D (15 min)

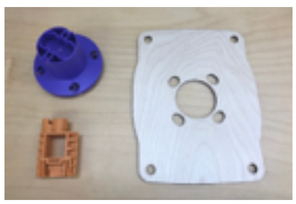
MATERIALS

The robotics kit referred to in this curriculum is the KIBO robotics kit, developed by the DevTech Research Group at Tufts University and made commercially available through KinderLab Robotics, Inc. (www.kinderlabrobotics.com). This curriculum uses the KIBO 21 kit, which includes the following:

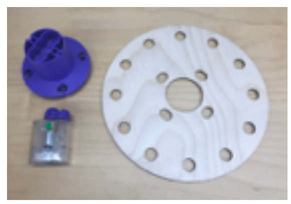
KIBO robot with wheels and motors

Input/output modules (distance, sound, and light sensors, lightbulb, sound recorder)

Expression module



Stage art platform



Rotating art stage with motor

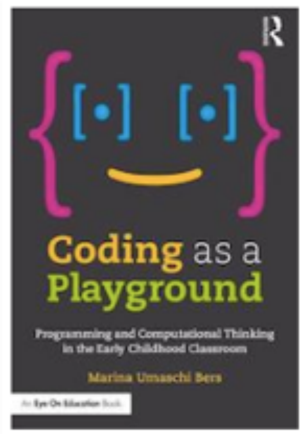
21 programming blocks

12 parameter cards

Other materials used in the curriculum are inexpensive crafts and recycled materials. The use of crafts and recycled materials, a practice already common in other domains of early childhood education, lets children build with a range of materials with which they are already comfortable with. There are many supplemental materials such as the KIBO Says cards and Activity Guide Cards that can be purchased through KinderLab Robotics (www.kinderlabrobotics.com). See Appendix A for the full list of materials for this curriculum.

PEDAGOGICAL FRAMEWORK: POSITIVE TECHNOLOGICAL DEVELOPMENT and DIALOGIC INSTRUCTION

The theoretical foundation of this curriculum, called **Positive Technological Development** (PTD), was developed by Prof. Marina Umaschi Bers and can be found in her books: *Blocks to Robotics: Learning with Technology in the Early Childhood Classroom* (Bers, 2008), *Designing Digital Experiences for Positive Youth Development: From Playpen to Playground* (Bers, 2012), and *Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom* (Bers, 2018). More information is included in the References section at the end of this curriculum.



The PTD framework guides the development, implementation and evaluation of educational programs that use new technologies to promote learning as an aspect of positive youth development. The PTD framework is a natural extension of the computer literacy and the technological fluency movements that have influenced the world of education but adds psychosocial and ethical components to the cognitive ones. From a theoretical perspective, PTD is an interdisciplinary approach that integrates ideas from the fields of computer-mediated communication, computer-supported collaborative learning, and the Constructionist theory of learning developed by Seymour Papert (1993) and views them in light of research in applied development science and positive youth development.

As a theoretical framework, PTD proposes six positive behaviors (six C's) that should be supported by educational programs that use new educational technologies, such as KIBO robotics. These are: **content creation, creativity, communication, collaboration, community building, and choices of conduct**. The six C's of PTD are highlighted in the activities throughout the curriculum with their respective icons:

CONTENT CREATION by designing a KIBO robot and programming its behaviors. The engineering design process of building and the computational thinking involved in programming foster competence in computer literacy and technological fluency. The use of Design Journals document for the children themselves, as well as for teachers and parents, their own thinking, their learning trajectories and the project's evolution over time.

CREATIVITY by making and programming personally meaningful projects, problem solving in creative playful ways and integrating different media such as robotics, motors, sensors, recyclable materials, arts and crafts, and a tangible programming language. Final KIBO projects that represent a theme found in the overall early childhood curriculum are a wonderful way to engage children in the creative process of learning.

COLLABORATION by engaging children in a learning environment that promotes working in teams, sharing resources and caring about each other while working with their KIBO robots. Collaboration is defined here as getting or giving help with a project, programming together, lending or borrowing materials, or working together on a common task. While working on their final KIBO projects, children create a collaboration web: a tool used to foster collaboration and support. Each child receives a printout with their photograph in the center of the page and the names and photographs of all the other children in the class arranged in a circle surrounding the central photo (see Appendix D for an example). Throughout the activity, with the teacher's prompting, each child draws a line from their own photo to the photos of the other children with whom they have collaborated. Children then write or draw "thank you cards" to the children with whom they have collaborated the most.

COMMUNICATION through mechanisms that promote a sense of connection between peers or with adults. For example, technology circles, when children stop their work, put their projects on the table or floor, and share their learning process. Technology circles present a good opportunity for problem solving as a community. Some teachers invite all the children to sit together in the rug area for this. It can also be helpful to make a "Robot Parking Lot" for all the robots to go while they are not being worked on, so children have empty hands and can focus at the technology circles. Each classroom will have its own routines and expectations around group discussions and circle times, so teachers are encouraged to adapt what already works in their class for the technology circles in this curriculum.

COMMUNITY BUILDING through scaffolded opportunities to form a learning community that promotes contribution of ideas. Final projects done by children are shared with the community via an open house, demo day, or exhibition. These open houses provide authentic opportunities for children to share and celebrate the process and tangible products of their learning with family and friends. Each child is given the opportunity not only to run their robot, but to play the role of teacher as they explain to their family how they built, programmed, and worked through problems.

CHOICES OF CONDUCT which provide children with the opportunity to experiment with "what if" questions and potential consequences, and to provoke examination of values and exploration of character traits while working with robotics. As a program developed following the PTD approach, the focus on learning about robotics is as important as helping children develop an inner compass to guide their actions in a just and responsible way.

In alignment with the Positive Technological Development (PTD) framework, this curriculum approaches literacy from the perspective of dialogic instruction. **Dialogic instruction** is a theory of learning (and teaching) premised on the belief that students engage with literacy instruction best when there are opportunities for them to engage in authentic, open-ended interpretation of texts. If a student does not have a voice, a position, or an evaluation of the text, then what good are literary skills? Only when she needs these tools for her own purpose, to help her achieve her own interpretation, and to convince others of it, will she have a reason and motivation (beyond getting a good grade) to acquire the tools being taught. This curriculum, in adherence with the theory of dialogic instruction, strives to place the student in the position of interpreter, with opportunities for authentic, open-ended interpretation of texts. This aligns with the curriculum's approach to coding where students are given opportunities for open-ended coding tasks that encourage them to explore their own expressive ideas.

CLASSROOM MANAGEMENT

Teaching robotics and programming in an early childhood setting requires careful planning and ongoing adjustments when it comes to classroom management issues. These issues are not new to the early childhood teacher, but they may play out differently during robotics activities because of the novelty and behavior of the materials themselves. Issues and solutions other than those described here may arise from classroom to classroom; teachers should find what works in their particular circumstances. In general, provide and teach a clear structure and set of expectations for using materials and for the routines of each part of the lessons (technology circles, clean up time, etc.). Make sure the students understand the goal(s) of each activity. Posters and visual aids can facilitate children's attempts to answer their own questions and recall new information. For example, teachers can use the mnemonic "KIBO" to introduce norms for playing with the KIBO robotics kit: **K**udos to…, **I** respect you, you respect me, **B**odies are safe, and **O**ops! Let's try it again.

GROUP SIZES

The curriculum refers to whole-group versus pair or individual work. In fact, some classrooms may benefit from other groupings. Whether individual work is feasible depends on the availability of supplies, which may be limited for a number of reasons. However, an effort should be made to allow students to work in as small groups as possible, even individually. At the same time, the curriculum includes numerous opportunities to promote conversations which are enriched by multiple voices, viewpoints, and experiences. Some classes may be able to have these discussions as a whole group. Other classes may want to break up into smaller groups to allow more children the opportunity to speak and to maintain focus. Some classes structure robotics time to fit into a "center time" in the schedule, in which students rotate through small stations around the room with different activities at each location. This format gives students more access to teachers when they have questions and lets teachers tailor instruction and feedback as well as assess each students' progress more easily than during whole-group work. It is important to find a structure and group size for each of the different activities (instruction, discussions, work on the challenges, and the final project) that meet the needs of the students and teachers in the class.

MANAGING MATERIALS

Classroom-scale robotics projects require a lot of parts and materials, and the question of how to manage them brings up several key issues that can support or hinder the success of the unit.

The first issue is accessibility of materials. Some teachers may choose to give a complete kit of materials to each child, pair, or table of several children. Children may label the kit with their name(s) and use the same kit for the duration of the curriculum. Other teachers may choose to take apart the kits and have materials sorted by type and place all the materials in a central location. Since different projects require different robotic and programming elements, this setup may allow children to take only what they need and leave other parts for children who need them. A word of caution, however: If materials are set up centrally, they must be readily visible and accessible, so children don't forget what is available to them or find it too much of a hassle to get what they need. Regardless, it is important to find a clearly visible place to set up materials for demonstrations, posters or visual aids to display for reference, and for robotics and programming materials for each lesson.

The second issue is usability. In some cases, children's desks or tables do not provide enough space to build a robot and program it. Care must be taken to ensure that children have enough space to use the materials available to them. If this is not the case, they may tend towards choosing materials that fit the space but not their robotics or programming goal. Teachers should carefully consider how to address these issues surrounding materials in a way that makes sense for their class's space, routines, and culture. Then, it is crucial to set expectations for how to use and treat materials. These issues are important not only in making the curriculum logistically easier to implement, but also because, as described in the Reggio Emilia tradition, the environment can act as the "third teacher" (Darragh, 2006).

ALIGNMENT OF ACADEMIC FRAMEWORK

This curriculum is designed for second grade and covers many foundational computer science and engineering skills. These academic frameworks are taught through a series of powerful ideas: algorithms, modularity, control structures, representation, hardware/software, design process, and debugging. Each powerful idea has activities and materials (in this case, the activities are tailored to fit the theme of *Where the Wild Things Are*) that encourage mastery of the powerful ideas from computational thinking (CT) and matches them with corresponding powerful ideas from literacy. This curriculum contains activities that specifically address the following literacy concepts and skills: the writing process, recalling, summarizing and sequencing, using foreshadowing, and using reading strategies such as predicting, summarizing, and evaluating.

Each lesson in this curriculum unit is aligned with standards from the **Common Core English Language Arts (ELA)/Literacy Framework**. The Common Core framework is "a set of standards that were created to ensure that all students graduate from high school with the skills and knowledge necessary to succeed in college, career, and life, regardless of where they live" (National Governors Association Center for Best Practices & Council of Chief State School Officers, 2010). Because Virginia is the first state to formally mandate K-12 computer science education, this curriculum is aligned with the **Virginia Department of Education's Standards of Learning for English and Standards of Learning for Computer Science** (Virginia Department of Education, 2017). Lessons in this curriculum are also aligned with nationally recognized computer science frameworks, including the **ISTE Standards for Students** (2017), **K–12 Computer Science Framework** (2016) and the **Massachusetts Digital Literacy and Computer Science (DLCS) Curriculum Framework** (2016).

Table 2: Alignment of Standards (on the next page)

	Powerful Ideas of Computational Thinking (CT) and Literacy Embedded in Each Lesson	Common Core ELA/ Literacy Framework (Grade 2)	Virginia English Standards of Learning (Grade 2)	Virginia Computer Science Standards of Learning (Grade 2)
1: Foundations	***CT***: *Design Process* ***Literacy***: *Writing Process*	**CCSS.ELA-LITERACY.W.2.5** With guidance and support from adults and peers, focus on a topic and strengthen writing as needed by revising and editing.	**Writing 2.12c** The student will expand writing to include descriptive detail. **Writing 2.12d** The student will revise writing for clarity.	**Algorithms and Programming 2.4** The student will plan and create a design document to illustrate thoughts, ideas, and stories in a sequential (step-by-step) manner. **Algorithms and Programming 2.6** The student will acknowledge that materials are created by others (e.g. author, illustrator, and website).
2: Technological Tools - Robots	***CT***: *Hardware/ Software, Representation* ***Literacy***: *Tools of Communication*	**CCSS.ELA-LITERACY.W.2.6** With guidance and support from adults, use a variety of digital tools to produce and publish writing, including in collaboration with peers.	**Writing 2.12c** The student will use available technology for reading and writing.	**Algorithms and Programming 2.5** The student will compare and contrast a group of items based on the attributes or actions of each item, with or without a computing device. **Computing Systems 2.7** The student will describe the characteristics of computing systems to include hardware, software, input, and output. **Computing Systems 2.8** The student will identify, using accurate terminology, simple hardware and software problems that may occur during use. **Impacts of Computing 2.13** The student will compare and contrast examples of how computing technology has changed and improved the way people live, work, and interact. **Networking and the Internet 2.15** The students will discuss in partners and as a class how information can be communicated electronically. **Data and Analysis 2.11** The student will construct and analyze data and organize it in a chart or graph in order to make a prediction, with or without a computing device.
	CT: *Hardware/ Software, Algorithms, Representation* ***Literacy***: *Summarizing/ Retelling the Sequence of a Story, Descriptive Language in Writing*	**CCSS.ELA-LITERACY.W.2.3** Write narratives in which they recount a well-elaborated event or short sequence of events, include details to describe actions, thoughts, and feelings, use	**Reading 2.5a** The student will use phonetic strategies when reading and spelling. S/he will use knowledge of consonants, consonant blends, and consonant digraphs to decode and spell words.	**Algorithms and Programming 2.1a** The student will construct step-by-step instructions both independently and collaboratively a) using sequencing.

3: Sequencing		temporal words to signal event order, and provide a sense of closure. **CCSS.ELA-LITERACY.RL.2.5** Describe the overall structure of a story, including describing how the beginning introduces the story and the ending concludes the action. **CCSS.ELA-LITERACY.RF.2.3.B** Know spelling-sound correspondences for additional common vowel teams.	**Reading 2.8h** The student will read and demonstrate comprehension of fictional texts. S/he will summarize stories and events with beginning, middle, and end in the correct sequence.	**Algorithms and Programming 2.2a** The student will construct programs to accomplish tasks as a means of creative expression using a block based programming language or unplugged activities, both independently and collaboratively
4. Programming	***CT**: Algorithms, Design Process* ***Literacy**: Descriptive Language in Writing*	**CCSS.ELA-LITERACY.W.2.2** Write informative/ explanatory texts in which they introduce a topic, use facts and definitions to develop points, and provide a concluding statement or section. **CCSS.ELA-LITERACY.W.2.3** Write narratives in which they recount a well-elaborated event or short sequence of events, include details to describe actions, thoughts, and feelings, use temporal words to signal event order, and provide a sense of closure.	**Oral Language 2.2c** The student will expand understanding and use of word meanings by clarifying and explaining words and ideas orally. **Oral Language 2.3d** The student will use oral communication skills by retelling information shared by others.	**Algorithms and Programming 2.1a** The student will construct step-by-step instructions both independently and collaboratively a) using sequencing. **Algorithms and Programming 2.2a** The student will construct programs to accomplish tasks as a means of creative expression using a block based programming language or unplugged activities, both independently and collaboratively a) using sequencing. **Algorithms and Programming 2.3** The student will analyze, correct, and improve (debug) an algorithm that includes sequencing and simple loops, with or without a computing device. **Computing Systems 2.8** The student will identify, using accurate terminology, simple hardware and software problems that may occur during use.
5. Debugging	***CT**: Debugging* ***Literacy**: Editing, Awareness of Audience*	**CCSS.ELA-LITERACY.RL.2.4** Describe how words and phrases (e.g., regular beats, alliteration, rhymes, repeated lines) supply rhythm and meaning in a story, poem, or song.	**Writing 2.12c** The student will expand writing to include descriptive detail. **Oral Language 2.2a** The student will expand understanding and use of word meanings by increasing listening and speaking vocabularies.	**Algorithms and Programming 2.1a and 2.1c** The student will construct step-by-step instructions both independently and collaboratively a) using sequencing and c) identifying events. **Algorithms and Programming 2.2a and 2.2c** The student will construct programs to accomplish tasks as a means of creative expression using a block based programming language or unplugged

				activities, both independently and collaboratively a) using sequencing and c) identifying events. **Algorithms and Programming 2.5** The student will compare and contrast a group of items based on the attributes or actions of each item, with or without a computing device.
6: Cause and Effect - Level 1	***CT***: *Control Structures, Representation, Sensors* ***Literacy***: *Spelling-Sound Correspondence*	**CCSS.ELA-LITERACY.RF.2.3** Know and apply grade-level phonics and word analysis skills in decoding words. (foundational skills)	**Oral Language 2.1** The student will demonstrate an understanding of oral language structure by creating and participating in oral dramatic activities. **Oral Language 2.4** The student will orally identify, produce, and manipulate various units of speech sounds within words.	**Algorithms and Programming 2.1a** The student will construct step-by-step instructions both independently and collaboratively a) using sequencing and c) identifying events. **Algorithms and Programming 2.2a and 2.2c** The student will construct programs to accomplish tasks as a means of creative expression using a block based programming language or unplugged activities, both independently and collaboratively a) using sequencing and c) identifying events.
7: Cause and Effect - Level 2	***CT***: *Algorithms, Modularity, Representation* ***Literacy***: *Descriptive Language in Writing*	**CCSS.ELA-LITERACY.RF.2.3.D** Decode words with common prefixes and suffixes. **CCSS.ELA-LITERACY.RL.2.4** Describe how words and phrases (e.g., regular beats, alliteration, rhymes, repeated lines) supply rhythm and meaning in a story, poem, or song.	**Reading 2.7b** The student will expand vocabulary when reading by using knowledge of prefixes and suffixes. **Reading 2.8e** The student will describe characters, setting, and important events in fiction and poetry.	**Algorithms and Programming 2.1a and 2.1c** The student will construct step-by-step instructions both independently and collaboratively a) using sequencing and c) identifying events. **Algorithms and Programming 2.2a and 2.2c** The student will construct programs to accomplish tasks as a means of creative expression using a block based programming language or unplugged activities, both independently and collaboratively a) using sequencing and c) identifying events.
8: Repeat Loops - Level 1	***CT***: *Control Structure, Modularity* ***Literacy***: *Repetition as a Literacy Device, Repetition in Word Forms*	**CCSS.ELA-LITERACY.RL.2.6** Acknowledge differences in the points of view of characters, including by speaking in a different voice for each character when reading dialogue aloud	**Reading 2.8j** The student will read and demonstrate comprehension of fictional texts by reading and rereading familiar stories, poems, and passages with fluency, accuracy, and meaningful expression.	**Algorithms and Programming 2.1b** The student will construct sets of step-by-step instructions (algorithms) both independently and collaboratively b) using loops. **Algorithms and Programming 2.2b** The student will construct programs to accomplish tasks as a means of creative expression using a block based programming language or unplugged activities, both independently and collaboratively b) using loops.

9: Repeat Loops - Level 2	***CT***: *Control Structures, Debugging, Sensors* ***Literacy***: *Descriptive Language, Perspectives in Narrative*		**Oral Language 2.3e and 2.3f** The student will use oral communication skills to follow three- and four-step directions and to give three- and four-step directions.	**Algorithms and Programming 2.1b and 2.1c** The student will construct sets of step-by-step instructions (algorithms) both independently and collaboratively b) using loops and c) identifying events. **Algorithms and Programming 2.2b and 2.2c** The student will construct programs to accomplish tasks as a means of creative expression using a block based programming language or unplugged activities, both independently and collaboratively b) using loops and c) identifying events.
10: If Statements	***CT***: *Control Structures, Debugging, Sensors* ***Literacy***: *Identify Conflict and Resolution, Making Predictions*	**CCSS.ELA-LITERACY.RL.2.3** Describe how characters in a story respond to major events and challenges.	**Reading 2.8a** The student will read and demonstrate comprehension of fictional texts by making and confirming predictions. **Reading 2.8c** The student will read and demonstrate comprehension of fictional texts by asking and answering questions about what is read.	**Algorithms and Programming 2.1c** The student will construct sets of step-by-step instructions (algorithms) both independently and collaboratively by c) identifying events. **Algorithms and Programming 2.2c** The student will construct programs to accomplish tasks as a means of creative expression using a block based programming language or unplugged activities, both independently and collaboratively by c) identifying events.
11: Final Project - Writing the Wild Rumpus Composition	***CT***: *Design Process* ***Literacy***: *Writing Process*	**CCSS.ELA-LITERACY.W.2.5** With guidance and support from adults and peers, focus on a topic and strengthen writing as needed by revising and editing.	**Oral Language 2.3a** The student will use oral (and written) language for different purposes: to inform, to persuade, to entertain, to clarify, and to respond. **Writing 2.12** The student will write stories, letters, and simple explanations. a) Generate ideas before writing. b) Organize writing to include a beginning, middle, and end for narrative and expository writing. c) Expand writing to include descriptive detail. d) Revise writing for clarity.	**Algorithms and Programming 2.1** The student will construct sets of step-by-step instructions (algorithms) both independently and collaboratively. **Algorithms and Programming 2.2** The student will construct programs to accomplish tasks as a means of creative expression using a block based programming language or unplugged activities, both independently and collaboratively. **Algorithms and Programming 2.3** The student will analyze, correct, and improve (debug) an algorithm that includes sequencing and simple loops, with or without a computing device **Algorithms and Programming 2.4** The student will plan and create a design document to illustrate thoughts, ideas, and stories in a sequential (step-by-step) manner.

12. Final Project - Coding the Wild Rumpus	***CT**: Design Process* ***Literacy**: Writing Process*	**CCSS.ELA-LITERACY.W.2.5** With guidance and support from adults and peers, focus on a topic and strengthen writing as needed by revising and editing.	**Writing 2.12d** The student will revise writing for clarity. **Writing 2.12** The student will write stories, letters, and simple explanations. a) Generate ideas before writing. b) Organize writing to include a beginning, middle, and end for narrative and expository writing. c) Expand writing to include descriptive detail. d) Revise writing for clarity. **Reading 2.8e** The student will describe characters, setting, and important events in fiction and poetry.	**Algorithms and Programming 2.1** The student will construct sets of step-by-step instructions (algorithms) both independently and collaboratively. **Algorithms and Programming 2.2** The student will construct programs to accomplish tasks as a means of creative expression using a block based programming language or unplugged activities, both independently and collaboratively. **Algorithms and Programming 2.3** The student will analyze, correct, and improve (debug) an algorithm that includes sequencing and simple loops, with or without a computing device. **Algorithms and Programming 2.4** The student will plan and create a design document to illustrate thoughts, ideas, and stories in a sequential (step-by-step) manner.

Lesson 1: Foundations

Powerful Idea From Computer Science:
Design Process

Powerful Idea From Literacy:
Writing Process

OVERVIEW

Students will learn about the Design Process and the Writing Process and understand how both processes are similar in nature but serve different purposes. Activities in this lesson encourage students to think and act like engineers and writers.

PURPOSE

While this lesson does not involve using the KIBO robotics kit, the activities set up an important foundation for how students engage in key computer science and literacy skills, such as brainstorming ideas, planning out a project, reviewing and revising ideas, and sharing ideas with peers.

ACTIVITIES

- What is an Engineer? (20 min)
- Engineers and Writers (10 min)
- Think Like an Engineer (10 min)
- How to Build a Robot (20 min)

STUDENTS WILL BE ABLE TO...

- Define engineer and understand that there are different types of engineers
- Compare and contrast the Design Process and Writing Process
- Use the Design and Writing Processes to design a robot

PREPARATION FOR TEACHERS

- ☐ Read through the Activity Guide
- ☐ Print pictures for What is an Engineer and Think Like an Engineer activities*
- ☐ Create anchor charts of the Design Process and Writing Process*
- ☐ Print Design Journals (one for each student - to be used throughout the entire unit)

MATERIALS

FOR THE TEACHER:

- 8-10 pictures of naturally occurring and man-made objects*
- Anchor chart of Design Process*
- Anchor chart of Writing Process*
- How-to book checklist

FOR STUDENTS:

- Design Journal (see Appendix C for example)

*See Appendix A for examples

VOCABULARY

- Cycle — something that moves in a circle (i.e. the seasons, a baseball field (compare to a football field that goes forward and backwards) the Design Process, the Writing Process)
- Design — a plan for a building or invention
- Engineer — someone who invents or improves things

Lesson 1: Activities

WHAT IS AN ENGINEER? (20 min)

Ask students: What do you think is an engineer? Do you know anyone who is an engineer? What kind of things do they do?

Explain to students that engineers do many different things, one of which is working with and designing computers and robots. In this lesson, they will learn about those kinds of engineers, but first, they need to understand what all engineers have to do: design. Introduce the steps of the Design Process.

An **engineer** is anyone who invents or improves things (for instance, just about any object you see around you) or processes (such as methods) to solve problems or meet needs. Any human-made object you encounter in your daily life was influenced by engineers. There are many different kinds of engineers including: biomedical engineers, aerospace engineers, computer engineers, and industrial engineers.

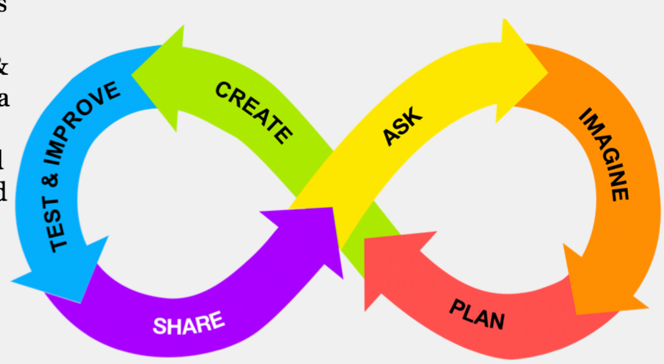
For descriptions and further activity ideas, check out the following resources:

- http://www.discovere.org/our-programs/engineers-week
- http://www.eie.org/eie-curriculum/curriculum-units
- *Engineering the ABC*'s by Patty O'Brien Novak

Design Process

When making projects, engineers follow a series of steps called the **Design Process**. It has 6 steps: ASK, IMAGINE, PLAN, CREATE, TEST & IMPROVE, and SHARE. The Design Process is a **cycle** – there's no official starting or ending point. You can begin at any step, move back and forth between steps, or repeat the cycle over and over!

Design Process song
(to the tune of "Twinkle, Twinkle")
Ask and imagine, plan and create,
Test and improve and share what we make.
(Repeat)



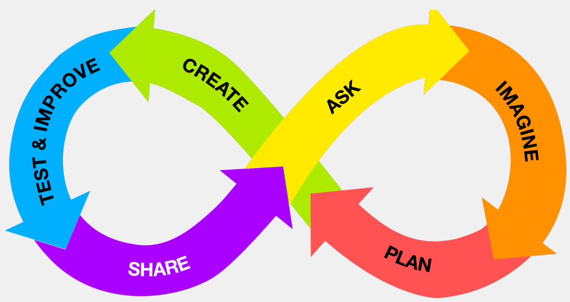
Show students a series of pictures of naturally occurring and man-made objects (show pictures one at a time). Examples of pictures are included in Appendix A. If students think that the object was built by an engineer, they should jump! If they think otherwise, they stay seated. Discuss students' reasoning. *Ask students: What made you think this was built by an engineer? What parts of the object made you think that way?*

ENGINEERS AND WRITERS (10 min)

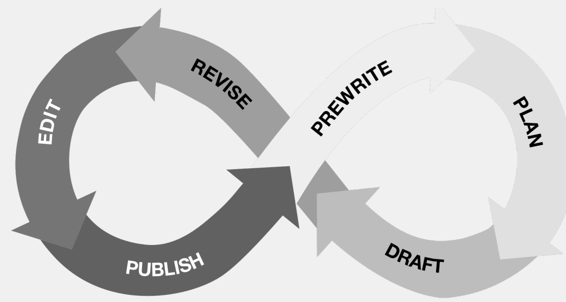
Show students the Design Process and the Writing Process side by side. Explain to students that both are creative processes that require imagination, planning, creating, revising, feedback, and sharing. Both engineers and writers turn ideas into projects that are shared with others. Ask students what other activities require a process (e.g., cooking, painting, getting good at a sport, etc.). Lead student-centered discussion on the similarities and differences between engineers and writers.

Writing Process

Just as engineers use the Engineering Design Process to design and create projects, writers use the Writing Process to brainstorm ideas, write a draft, make revisions, and share their writing with others. The Writing Process is also a **cycle** – there's no official starting or ending point, and you can move back and forth between steps!



Design Process

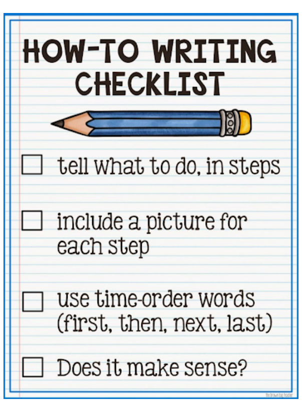


Writing Process

THINK LIKE AN ENGINEER (10 min)

Explain to students that everyone in the class is going to start thinking like an engineer! *Ask students: Have you seen or interacted with robots before? What do they look like? What kinds of different parts make up a robot? How do you think engineers build robots? ? What might happen if the engineers went straight to building a robot without drawing out a plan first?* The purpose of this activity is to engage students in thinking about design and how engineers use different types of materials to create their products.

HOW-TO-BOOKS: BUILDING A ROBOT (20 min)



How-to-Books are a low-stress entry point into writing. After all, all students know how to do something and the structure of a how-to book is fairly simple. In addition, pictures can easily take the place of words. We even suggest that each step in a how-to book should be accompanied by a sketch or picture.

Pass out the Design Journals. Ask students to create a "How-To Book" for building their own robot. Ask students to include specific details so that someone else can learn how to build their robot simply by reading these instructions. Depending on the students' writing level, this activity may need more framing. A wonderful resource for How-To-Books can be found at: https://www.education.com/lesson-plan/creating-a-how-to-book/. Students will share their How-To books in pairs in a later lesson.

Lesson 2: Technological Tools - Robots

Powerful Idea From Computer Science:
Hardware/Software, Representation

Powerful Idea From Literacy:
Tools of Communication

OVERVIEW

The advancement of technology over the years has changed the way people communicate and do things. In this lesson, students will begin to understand how technology and communication tools have evolved. Students share ideas, learn about the different characteristics of robots, and learn about KIBO's programming language.

PURPOSE

By learning to code with the KIBO programming blocks, students understand how programming languages are different from natural spoken languages. Both require clear and precise communication, but while humans can understand many different types of genres of speech, KIBO can only understand commands. Furthermore, understanding that robots have special parts (hardware) to let them follow instructions (software) is a powerful idea of computational thinking, which will help students build more complex programs in subsequent lessons.

ACTIVITIES

- Robot Corners (15 min)
- Characteristics of Robots (10 min)
- Tools of Communication (10 min)
- Human Language vs. Code Language (10 min)
- KIBO Says (15 min)

STUDENTS WILL BE ABLE TO...

- Identify characteristics of a robot
- Compare human languages and programming languages
- Create a simple algorithm using the KIBO programming blocks

PREPARATION FOR TEACHERS

- ☐ Read through the Activity Guide
- ☐ Print pictures for Robot Corners activity and the messages for Tools of Communication activity*
- ☐ Create anchor chart for the Characteristics of Robots activity*
- ☐ Go through the KIBO Says cards and take out only the blocks listed in the Materials section

MATERIALS

FOR THE TEACHER:

- 1 piece of blank chart paper for the Characteristics of Robots activity*
- 8-10 pictures of robots and non-robots*
- Handwritten and typed message*

FOR STUDENTS:

- Design Journal (see Appendix C for example)
- KIBO Says cards: Begin and End blocks, blue Motion blocks

*See Appendix A for examples

VOCABULARY

- Robot — a machine that can be programmed to do different things
- Barcode — a pattern of lines that are readable by machines (like the KIBO robot)
- Program — a set of instructions for a robot

Lesson 2: Activity

ROBOT CORNERS (15 min)

As explained in the book *Blocks to Robots* by Dr. Marina Bers (2008, p. 70), **robots** can "refer to a wide range of machines...that take on different forms... and can perform autonomous or preprogrammed tasks". Despite their differences, all robots are "capable of movement under some form of control and can be used to perform physical tasks." For example, you can give the robot a set of instructions for its motors in order to make the robot move. The robotic "brain", just like the human brain, has the programmed instructions that make the robot perform its behaviors. It may be helpful to watch video clips of different types of robots in action such as home robots, space robots, factory robots, hospital robots, and child-made robots.

Ask all students to stand in a line or circle where they can see you. Designate three corners of the classroom: one corner for "Robots", one corner for "Maybe Robots", and one corner for "Not Robots." One at a time, show a variety of different pictures of robots and non-robots (e.g. computers, cars, animals, foods, famous robots such as Wall-E and R2D2). Ask students to move to the corner that they think represents the picture. Then ask a few students to explain why they think the picture is a robot or not a robot or why they think it might be a robot. Do not reveal answers until after the next activity: Characteristics of Robots. It is important in this activity for students to share their ideas about they think a robot is.

CHARACTERISTICS OF ROBOTS (10 min)

Read the true/false statements about robots below. Ask students to stand (or make another movement like snapping or waving their fingers in the air) for statements they think are true and sit down for statements they think are false.

Extended Graphing Activity: As you go along, make a graph on a piece of chart paper with True and False for each question along the horizontal axis and number of students along the vertical axis. Have students place a marker (sticker, symbol, etc.) in the "True" or "False" column. Explain to students that the graph allows us to see whether there were more "True" or "False" responses for each question.

1. *Robots are machines (TRUE).*
2. *All robots are made of the same materials (FALSE).*
3. *Robots must have moving parts (TRUE).*
4. *Robots can think by themselves (FALSE).*
5. *All robots look alike (FALSE).*
6. *Robots must be able to move around the room (FALSE).*
7. *Robots are operated using remote controls (FALSE).*
8. *People tell robots how to behave with a list of instructions called a program (TRUE).*
9. *Some robots can tell what is going on around them (TRUE)*
   (Examples: sensing light, temperature, sound, or a touch.)
10. *Robots are alive (FALSE).*

Choose 1-2 pictures from the Robot Corners activity and lead student-centered discussion about why that picture represents a robot or is not a robot based on what they have just learned about robots.

For further activity ideas on robots, check out the following resources:

- *Robots, Robots Everywhere!* by Sue Fliess
- *National Geographic Readers: Robots* by Melissa Stewart

TOOLS OF COMMUNICATION (10 min)

Have students sit in a circle and play a game of "Telephone", in which one student thinks of a message and whispers it to the person sitting next to them, who then whispers to the person next to them, and so on and so forth until the message gets to the last person. Ask the last person and the first person to say their messages out loud and compare the two messages. *Ask students: Were the two messages the same? Why or why not? What are some other ways we could use to pass along a message?*

Repeat the game one final time, this time by giving each student a typed and printed version of the message. Have a few students read out their printed message. *Ask students: How was this better than the last two rounds? Are all students able to receive the same information? (Yes)*

Repeat the game one final time, this time by giving each student a typed and printed version of the message. Have a few students read out their printed message. *Ask students: How was this better than the last two rounds? Are all students able to receive the same information? (Yes)*

At the end of the activity, explain to students how this mirrors the evolution of writing technology from oral societies to scribal writing to post-printing press. Help students draw the connection to the evolution of computers and robotic technologies. More specifically, explain to students that if we had to program robots without writing, it would be messy, but we can use computer writing to program robots, and that is called **code**.

HUMAN LANGUAGE VS. CODE LANGUAGE (10 min)

This activity also has two parts: Meaning of Words and KIBO's Language. Both activities serve to illustrate how human languages (written and spoken) can be used to communicate a variety of things (e.g. sarcasm, allusions, hyperbole/ exaggerations, etc.), whereas programming languages are more structured and literal.

For the Meaning of Words activity, the goal is to remind students of what Mikhail Bakhtin calls, "heteroglosia," the multiple meanings we all carry for each word. In simple terms, human language is much more dynamic than code language. Ask students what people actually mean when they say certain things. For example:

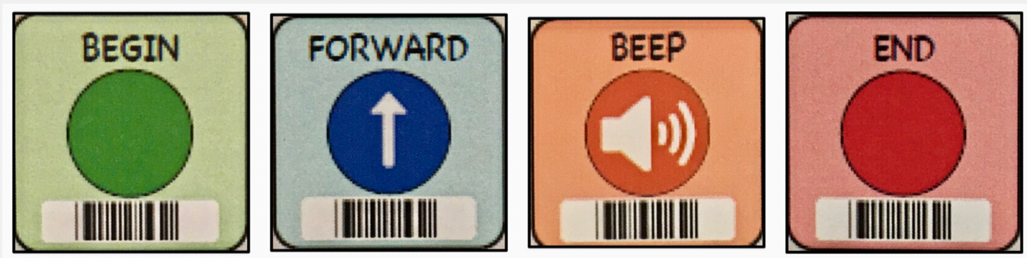
> *I'm so hungry I could eat a horse!*
> *I have a million things to do today.*
> *My homework is taking forever to get done.*

For the KIBO's Language activity, show students the large KIBO Says cards. Have students point out what they see on each block: the text, the icon, colors, the barcode, etc. *Ask students: What part of the block is KIBO's language? Is it the words, or the pictures, or something else?* Once students identify the barcode as the answer, discuss other objects or places where they have encountered barcodes.

Then *ask students: Do you think KIBO can think on its own? Can KIBO make its own program?* Lead student-centered discussion on how robots are programmed by humans and cannot think for themselves. Everything that KIBO says and does is determined by how the programmer chooses the program, or set of instructions, for KIBO. For example, we say we want KIBO to move forward, but KIBO reads the barcodes for the Begin, Forward, and End blocks.

What is a Program?

A **program** is a sequence of instructions that the robot acts in order. Each instruction has a specific meaning, and the order of the instructions affects the robot's overall actions. This is an example of a KIBO program.



KIBO SAYS (15 min)

In order to program the KIBO robot, students first need to learn KIBO's language: the programming blocks! This activity is played like the traditional "Simon Says" game, in which students repeat an action if Simon says to do something. Briefly introduce each programming instruction and what it means (use only the blocks listed in the Materials section in this lesson).

Have the class stand up. Hold up one big KIBO icon at a time and say, "Programmer says to __________". Go through each individual instruction a few times until the class seems to get it. Once students are familiar with each instruction, ask for volunteers to be the Programmer who gives the class full programs to run through (e.g. Begin, Spin, Forward, End). Just like in the real "Simon Says" game, the Programmer can try to be tricky! For example, if the Programmer forgets to give a Begin or End instruction, should the class still move? Just like Simon Says, if the Programmer forgets to say, "Programmer says to ____________", then students should sit down! This will help reinforce the concept that KIBO is programmed by humans.

Lesson 3: Sequencing

Powerful Idea From Computer Science:
Hardware/Software, Algorithms, Representation

Powerful Idea From Literacy:
Summarizing/Retelling the Sequence of a Story, Descriptive Language in Writing

OVERVIEW

Students will learn about sequencing in programming and think about how it relates to sequencing in literacy, and why order matters in both cases. Once students become familiar with some of the KIBO programming blocks, they will learn about the different parts of the KIBO robot.

PURPOSE

In the previous lesson, students began learning about different KIBO blocks. Now they will engage in goal-oriented programming, in which students purposefully choose actions in a specific order to achieve a particular outcome. Understanding that order matters is an important skill for students not only in computer science and literacy, but also in their everyday lives as they learn to tie their shoelaces, reflect on the day's activities, plan a family vacation, and more.

ACTIVITIES

- Where the Wild Things Are (20 min)
- Order Matters (15 min)
- Program the Teacher with KIBO Blocks (10 min)
- Meet the KIBO Robot (15 min)

STUDENTS WILL BE ABLE TO...

- Understand why order matters when programming a robot or telling a story
- Identify the different parts of the KIBO robot

PREPARATION FOR TEACHERS

- ☐ Read through the Activity Guide
- ☐ Go through the KIBO Says cards and take out only the blocks listed in the Materials section
- ☐ Print large letter cards*
- ☐ Ensure all KIBO bodies have 4 working AA batteries
- ☐ Sort KIBO blocks and pieces (listed in Materials section) by part and place in a central location

MATERIALS

FOR THE TEACHER:

- 1 copy of Where the Wild Things Are by Maurice Sendak
- Large letter cards: A, R, C*
- Large KIBO Says cards: Begin and End blocks, blue Motion blocks, Beep and Sing blocks
- 1 flathead screwdriver
- Extra AA batteries

FOR STUDENTS:

- Design Journal (see Appendix C for example)
- KIBO bodies, wheels, motors, and art platforms

*See Appendix A for examples

VOCABULARY

- Instruction – a direction that a robot will understand
- Order – parts of a group arranged in a specific way (e.g., smallest to largest, tallest to shortest)
- Program – a complete set of instructions for a robot
- Scanner – electronic device for reading printed barcodes
- Sequence – the order of instructions that a robot will follow exactly (often used interchangeably with algorithm)
- Main board – the robot's "brain" that has the programmed instructions that the robot to perform its behaviors
- Motor – the part of a robot that makes it move
- Wheels – the round parts of a vehicle that turn in circles and allow it to move

Lesson 3: Activities

WHERE THE WILD THINGS ARE (20 min)

Read the book *Where the Wild Things* Are as a class; if needed, read the book a second time. Lead a student-centered discussion that reviews the events of the story. You can prompt the students: Who can summarize the main events in this story? (e.g. first he made some mischief, then more, then yelled at his mother, etc.). *Then ask students: What if the first scene was Max on a boat? How would that change the story? What about if Max had smelled the food before making more mischief?* The purpose of this activity is to get students to think about sequencing in narrative.

ORDER MATTERS (15 min)

This activity has two parts: Guide a Friend and Rearrange Letters. The purpose of these activities is to reflect on the importance of sequencing both in computer science and literacy.

For the Guide a Friend activity, divide your students in pairs. Ask Partner A to write instructions in their Design Journals that tells Partner B how to get to a specific place in the room. Then ask Partner B to read and perform those instructions. If they do not reach the correct location, ask Partner A to revise their instructions, and ask Partner B to try again. For example:

Partner A writes to Partner B:	1. To get to the door, walk forward 10 steps. 2. Turn around to face the window. 3. Walk 5 more steps. 4. Stop when you reach the door
Partner B reads and performs the instructions but ends up facing a wall.	
Partner A revises instructions	1. To get to the door, walk forward 7 steps. 2. Turn around to face the window. 3. Walk 5 more steps. 4. Stop when you reach the door
Partner B tries again and reaches the door!	

Ask students: Were you able to get to the correct location? What instruction was confusing or led you in the wrong direction? How could that be corrected?

For the Rearrange Letters activity, ask three students to volunteer to hold one of the three large letter cards: A, R, and C. Ask the three students to spell "A-R-C" by arranging themselves in a line. Then ask the three students to spell the word "C-A-R" by rearranging themselves. *Ask the class: What changed when the three volunteers moved their positions? Do the two words mean the same or different things?* Explain to students that letters are symbols for sounds and are strung together in different ways to make different words. When the position of the letters changed, the way we sounded out the letters and the word itself (hence the meaning of the word) also changed.

Conclude the activity by reflecting on the importance of sequencing in literacy and computer science. *Ask students: Why did the order matter in each activity?*

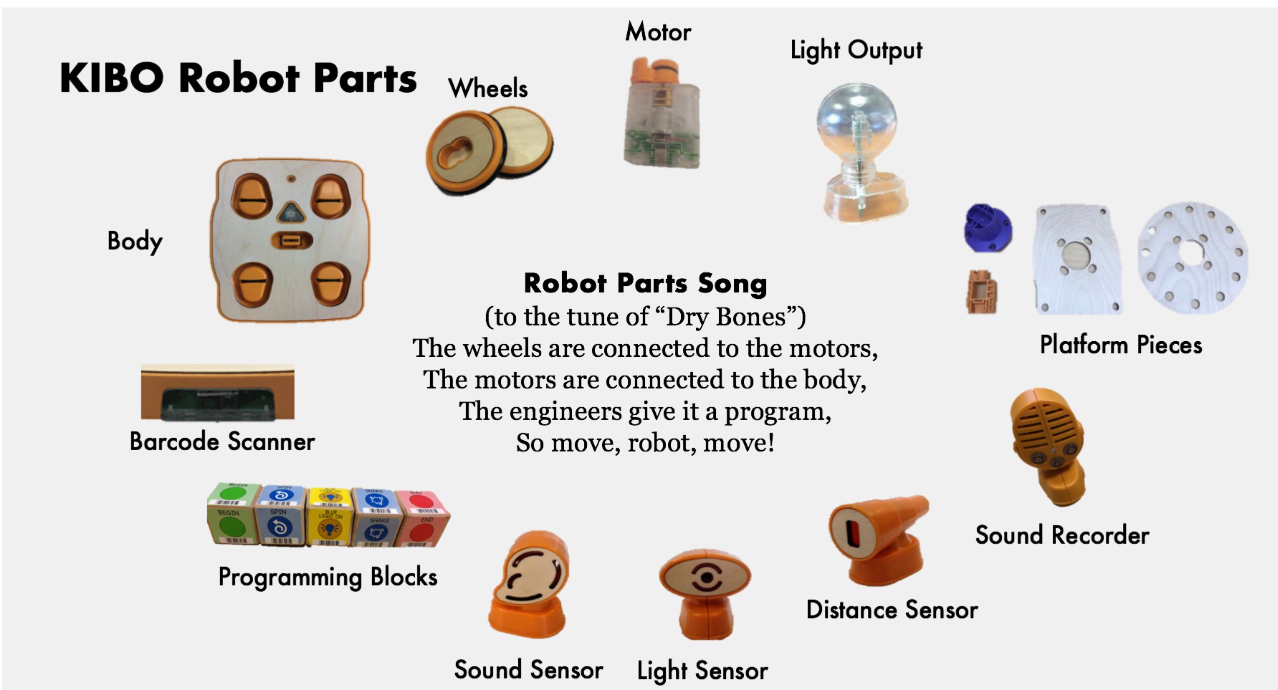
PROGRAM THE TEACHER WITH KIBO BLOCKS (10 min)

Using the KIBO Says cards, students will work together as a class to "program" their teacher to move from one part of the room to the other. Be silly! An example would be for the students to "program" their teacher to move from the front of the room to the library area by using these blocks: Begin, Forward, Spin, Turn Left, Forward, Forward, End. The goal of this game is for students to practice sequencing as a class before working individually or in their small groups. Before the teacher-robot moves, students can make predictions about where the teacher-robot will end up. It may be helpful to let the students make mistakes in order to foster a discussion on sequencing and debugging.

MEET THE KIBO ROBOT (15 min)

Take out KIBOs and blocks. Explain to students that today they will be learning how to put together the different parts of the KIBO robot. Show students a KIBO robot body. *Ask students: What parts do you see through the clear backside of KIBO? What do you think those parts do? What do the batteries do? What are some other objects you have seen that have the same function? (e.g. KIBO's wheels are like the tires on a car)*

Using the KIBO parts guide below, introduce the KIBO robot's key parts and their functions. Teach the "Robot Parts Song" and have students sing and dance along. Explain to students that the song helps us understand how to put the KIBO robot together. Demonstrate how to attach the wheels, motors, and art platforms. If time permits, allow students to work in pairs to assemble their own KIBO robot.



Lesson 4: Programming

Powerful Idea From Computer Science:
Algorithms, Design Process

Powerful Idea From Literacy:
Descriptive Language in Writing

OVERVIEW

Students will learn about sequencing in programming and think about how it relates to sequencing in literacy. Students will program KIBO to dance the Hokey-Pokey, or if you wish, a different children's song where students can program a robot to dance to the words. At the end of the lesson, students will demonstrate their current level of understanding by completing the first Solve-It assessment.

PURPOSE

In the previous lesson, students had the opportunity to engage with KIBO's hardware and software separately. Now they will engage in goal-oriented programming, in which students purposefully choose their KIBO blocks and place them in a specific order to achieve a particular outcome.

ACTIVITIES

- Dance the Hokey-Pokey (5 min)
- Program the Hokey-Pokey (20 min)
- Hokey-Pokey Reflection (10 min)
- Share Creations (10 min)
- Solve-It Assessment A (15 min)

STUDENTS WILL BE ABLE TO...

- Tell and retell a story clearly and effectively
- Identify common errors with scanning KIBO programs and troubleshoot them
- Practice scanning programs with KIBO
- Learn strategies for debugging and editing

PREPARATION FOR TEACHERS

- ☐ Read through the Activity Guide
- ☐ Ensure all KIBO bodies have 4 working AA batteries
- ☐ Sort KIBO blocks and pieces (listed in Materials section) by part and place in a central location
- ☐ Print Solve-It Assessment A (one for each student)

MATERIALS

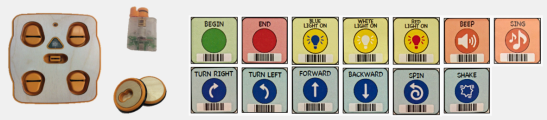
FOR THE TEACHER:

- Anchor chart of discussion sentence starters*
- 1 flathead screwdriver
- Extra AA batteries

FOR STUDENTS:

- Design Journal (see Appendix C for example)
- KIBO bodies, wheels, motors, Begin and End blocks, blue Motion blocks, yellow Light blocks, Beep and Sing blocks
- KIBO stickers

*See Appendix A for examples



VOCABULARY

- Instruction – a direction that a robot will follow
- Program — a complete set of instructions for a robot
- Scanner — electronic device for reading printed barcodes

Lesson 4: Activities

DANCE THE HOKEY-POKEY (5 min)

Explain to students that today they will program KIBO to do the Hokey-Pokey. Sing and dance the Hokey Pokey as a class to make sure everyone knows and remembers it. Conclude with a “robot verse”:

You put your right hand in,
You put your right hand out,
You put your right hand in,
And you shake it all about,

You do the hokey pokey
and you turn yourself around
That what it's all about. (clap, clap!)

2) left hand
3) right foot
4) left foot
5) head
6) whole self

You put your robot in, you put your robot out,
You put your robot in, and you shake it all about.
You do the Hokey Pokey, and you turn yourself around.
And that’s what it’s all about. (Clap, clap.)

PROGRAM THE HOKEY-POKEY (20 min)

Take out KIBOs and blocks. Remind students how to assemble the KIBO blocks and scan a complete program with KIBO. Have several students share out their strategies for scanning KIBO. Individually or in pairs, students program their KIBOs to do the Hokey-Pokey.

HOKEY-POKEY REFLECTION (10 min)

In their Design Journals, ask students to record their Hokey-Pokey programs by using the KIBO stickers to write out the blocks in their program. *Ask students: How many times did you use each programming block? What order did you put the blocks in? Why did you choose this particular order?* Have students share out the number of times they used the Forward block or the Sing block. *Ask students: Did the whole class use the same number of each block?*

SHARE CREATIONS (10 min)

When all groups are done with their Hokey-Pokey robot programs, ask the whole class to play their programs at once and dance the Hokey-Pokey! This is the first time that students engage in goal-oriented programming. Using the Discussion Sentence Starters anchor chart, ask students about their challenges of programming: *What problems did you have when you were scanning blocks? Did you ever get an error message? Did you ever feel frustrated or disappointed? Why did you feel that way?* Note down students’ responses on a piece of paper so that you can come back to these points in the next lesson.

SOLVE-IT ASSESSMENT A (15 min)

On the Appendix B-1 you will find assessment A. Please hand out one assessment sheet to every child in your class.

Instructions:

- Read each question and option out loud to the group. Students can ask to have questions or options read out loud up to 3 times.
- Instruct children to circle only 1 answer per question.
- Make sure students answer the questions by themselves. Students should not be discussing or copying answers.
- Hand in completed answer sheets to Angela de Mik or a member of your school's assessment team.

Lesson 5: Debugging

Powerful Idea From Computer Science:
Debugging

Powerful Idea From Literacy:
Editing, Awareness of Audience

OVERVIEW

In this lesson, students learn the importance of communicating effectively to an audience. Students engage in this learning by retelling a story to their peers and "edit" their story when their audience is confused and needs more clarification. Students connect this idea to when the KIBO robot does not perform the intended instructions. The process of figuring out what went wrong and how to fix things is called debugging.

PURPOSE

The parallel of editing in literacy and debugging in computer science is crucial to students' understanding of the differences between humans and computers/robots. Humans might be able to tell what a storyteller is trying to communicate even if they leave out a few details; however, a computer is far less flexible. Furthermore, this lesson allows students to not only encounter obstacles, but also to identify and troubleshoot these issues, thus building their confidence to tackle later, more challenging lessons.

ACTIVITIES

- Tell a Story (15 min)
- Why is KIBO Confused? (15 min)
- Free Play (20 min)
- Debugging Reflection (10 min)

STUDENTS WILL BE ABLE TO…

- Identify common errors with scanning KIBO programs and troubleshoot them
- Practice scanning programs with KIBO
- Learn strategies for debugging and editing

PREPARATION FOR TEACHERS

- ☐ Read through the Activity Guide
- ☐ Prepare Why is KIBO Confused? anchor chart
- ☐ Ensure all KIBO bodies have 4 working AA batteries
- ☐ Sort KIBO blocks and pieces (listed in Materials section) by part and place in a central location
- ☐ Make copies of students' How-To-Book drafts from Lesson 1

MATERIALS

FOR THE TEACHER:

- Why is KIBO Confused? anchor chart
- 1 flathead screwdriver
- Extra AA batteries

FOR STUDENTS:

- Design Journal (see Appendix C for example)
- KIBO bodies, wheels, motors, Begin and End blocks, blue Motion blocks, yellow Light blocks, Beep and Sing blocks

*See Appendix A for examples

VOCABULARY

- Debug — to find and solve a problem in a computer program
- Edit — to make changes to something

Lesson 5 Activities

TELL A STORY (15 min)

Framing for the Teacher: Even the most basic forms of writing (letter and letter-like forms) require a high level of abstraction that speech does not. Education psychologist, Lev Vygotsky explained, “In learning to write, the child must disengage himself from the sensory aspects of speech and replace words by images of words” (Vygotsky, 2012, p.181). Writing requires symbolization of the sound image into written signs (letters, syllables, etc.). It is this abstract quality of written language, specifically double abstraction- abstraction from the sound of speech and abstraction from the interlocutor (the reader)- that makes it challenging. Of course, second graders are at the very beginning of this journey. **The goal at this stage of development is to understand that even though all the details exist in your head, if you don’t provide them for your reader, your story won’t make sense. This is the debugging/editing challenge for students at this stage of development.**

Prior to the start of this lesson (the day or night before), make copies of the students’ How-To-Book drafts in their design journals. Split the class into pairs, and ask pairs to trade their How-To-Book drafts and read them. Have each partner try to explain to the other person how to create the boat, as they understood it, according to their partner’s How-To-Book. This activity can be fun and light. The big idea is that it can be hard to communicate what’s in our head to someone or something else.

Explain to students that with writing there may be multiple ways to communicate the same thing, and even if we misspell a word or make a grammatical error, our message may still be clear. However, when robots or computers are the audience, we have to make sure to communicate in the way that the machines understand. There is much less margin for error.

WHY IS KIBO CONFUSED? (20 min)

In Lesson 3, students shared challenges of scanning KIBO blocks and other issues that they experienced while creating their Hokey-Hokey programs. Check back on your notes from that discussion and prepare an anchor chart noting 4-5 of these challenges on the left side of the chart, leaving the right side empty for students to provide solutions in this activity.

Present the anchor chart to students. Explain to students how in the previous lesson, students encountered different challenges, such as not being able to scan the blocks properly, seeing a red light or hearing a minor key sound when scanning the blocks, etc. Other examples of common errors can be found in this KIBO troubleshooting tip sheet: http://kinderlabrobotics.com/wp-content/uploads/2017/10/KIBO-10-Quick-Start-Guide.pdf.

Ask students to brainstorm 1-2 solutions for every solution. An example is provided below:

Challenge #1: It’s hard to scan the blocks.	Solution #1: Separate the blocks instead of connecting the pegs. Scan each block individually. Solution #2: Ask your partner to cover the other barcodes on the left and right of the block you’re trying to scan.
Challenge #2: When I accidentally scan the End block twice, it gives me a red light, and I have to scan the program all over again.	Solution #1: Tilt the KIBO immediately after scanning the block so that the barcode scanner doesn’t accidentally scan it twice.

Explain to students that **debugging** is a method used to understand how to fix things when engineers program robots, and the robots do not work. By identifying these problems and different solutions to solve them, students are debugging.

Debugging is a word used in computer science to describe when people find errors in their computer programs and use different strategies to solve the problem. While the word "bug" was used in other scientific fields, the word "debugging" is attributed to Admiral Grace Hopper, who back in the 1940s found a moth stuck inside the computer (computers used to be that big!), which caused an error in the system. She was able to resolve the error by taking out the bug, hence the word "debugging"!

For further activity ideas and examples of pictures, check out the following resources:

- https://www.computerhope.com/issues/ch000984.htm
- https://www.npr.org/sections/alltechconsidered/2015/11/23/457129179/the-future-of-nanotechnology-and-computers-so-small-you-can-swallow-them

FREE PLAY (20 min)

Take out KIBOs and blocks. This activity is a great opportunity for students to freely explore with the KIBO robot and the programming blocks. Encourage students to try to make these mistakes purposefully and to practice debugging! By the end of this activity, students should feel comfortable scanning a complete program onto KIBO.

DEBUGGING REFLECTION (10 min)

Pass out students' Design Journals. Ask students to reflect about one of the problems they had with KIBO. *What was the problem?* Ask students to explain why KIBO wasn't understanding what they wanted KIBO to do. *How did you change the way you scanned (communicated) so that KIBO would understand?* Students can reflect in their Design Journals by drawing a picture of how they debugged, or if they can, write about their problem solving strategy.

Lesson 6: Cause and Effect - Level 1

Powerful Idea From Computer Science:
Control Structures, Representation, Sensors

Powerful Idea From Literacy:
Spelling-Sound Correspondence

OVERVIEW

In this lesson, students will learn about cause and effect and sensors by being introduced to two new modules: the Sound Sensor and the Sound Recorder. The Sound Sensor uses an event (wait for clap) before performing the subsequent action. The Sound Recorder allows students to record up to three different sound clips, depending on the shape of the recorder button they press (circle, square, triangle). Students will learn that for KIBO to play their sound clip, they must use the orange Sound Recorder block that has the same shape (circle, square, triangle).

PURPOSE

When students learned to program with the Beep and Sound blocks in previous lessons, KIBO did not require a sound sensor because those sounds were produced from the robot (output). In this lesson, students learn how robots can take in information from the environment to then perform an action (input). These concepts are integral to the understanding of control structures, which will prove useful in subsequent lessons.

ACTIVITIES

- What did Max Sense (15 min)
- KIBO Sound Sensor (5 min)
- Shape Shifting (15 min)
- KIBO Sound Recorder (5 min)
- Free Play (15 min)
- Solve-It Assessment B (10 min)

STUDENTS WILL BE ABLE TO...

- Distinguish between human senses and robot sensors
- Use the KIBO Sound Sensor with its appropriate Wait for Clap block
- Record a sound clip successfully using the Sound Recorder module and Sound Recorder blocks

PREPARATION FOR TEACHERS

- ☐ Read through the Activity Guide
- ☐ Ensure all KIBO bodies have 4 working AA batteries
- ☐ Sort KIBO blocks and pieces (listed in Materials section) by part and place in a central location
- ☐ Print Solve-It Assessment B (one for each student)

MATERIALS

FOR THE TEACHER:

- 1 copy of *Where the Wild Things Are* by Maurice Sendak
- KIBO Says cards: orange Sound Recorder blocks
- 1 flathead screwdriver
- Extra AA batteries

FOR STUDENTS:

- Design Journal (see Appendix C for example)
- Construction paper and markers
- KIBO bodies, wheels, motors, Begin and End blocks, blue Motion blocks, yellow Light blocks, Beep and Sing blocks, Wait for Clap blocks, Sound sensors, Sound Recorder module, orange Sound Recorder blocks

VOCABULARY

- Senses — the way humans and animals take in information about the surrounding environment. Humans have five senses: touch, taste, smell, sight, and hearing
- Sensor — a special part that helps machines take in information about the surrounding environment; there are sensors that are very much like human senses
- Event — an action that causes something to happen
- Circle — a round, closed shape with no edges
- Record — to make something (like a sound) permanent so that it can be played back at a later time
- Sound — a type of energy made by vibrations in the air that we can hear
- Square — a closed shape with four equal sides
- Triangle — a closed shape with three sides

Lesson 6: Activities

WHAT DID MAX SENSE? (15 min)

Throughout the *Where the Wild Things Are* story, Max uses his five senses: taste, smell, touch, hearing, and sight. *Ask students: What body parts do humans use to sense things in our environment?* As a class, decide on a movement that represents each of the five senses. For example, you might decide to point to your tongue for taste, nose for smell, fingers for touch, ear for hearing, and eyes for sight.

At your discretion, reread the story as a class, or select pages from the story. As you read, pause at different points and ask students to do the movement that corresponds to what Max is sensing. This is an important chance to point out to students that literature "shows" instead of "tells." For example, when we read, "The wild things roared their terrible roars and gnashed their terrible teeth and rolled their terrible eyes and showed their terrible claws," we understand what exactly Max was seeing (and maybe even smelling if they had bad breath with those terrible teeth!). Explain to students how literature sometimes uses "poetic" language, whereas computer science uses literal and command-oriented code language.

Below are examples of quotes from the story to pause and have students identify Max's senses:

Taste	*"So he was sent to bed **without eating anything**."*
Smell	*"Then all around from far away, across the world, he **smelled good things to eat**, so he gave up being King of where all the Wild Things are"*
Touch	*"The night Max wore his **wolf suit** and made mischief of one kind and another"* *"and into the night of his very own room where he found his supper waiting for him and **it was still hot**."*
Hearing	*"His mother **called him 'WILD THING'**"* *"And they were frightened and **called him the most Wild Thing of all** and made him King of all Wild Things"* *"But the **wild things cried**, "Oh please don't go we'll eat you up-we love you so!" And **Max said**, "No!"* *"The wild things **roared their terrible roars** and gnashed their terrible teeth and rolled their terrible eyes and showed their terrible claws, but Max stepped into his private boat and waved goodbye"*
Sight	*"That very night in Max's room, **a forest grew**..."* *"When he came to the place where the Wild Things are...**terrible claws**"* *"Till Max said BE STILL...**blinking once**"* *"The wild things roared their terrible roars and **gnashed their terrible teeth and rolled their terrible eyes and showed their terrible claws**, but Max stepped into his private boat and **waved goodbye**"* *"and into the night of his very own room where he **found his supper waiting for him** and it was still hot."*

KIBO SOUND SENSOR (5 min)

Take out KIBOs and blocks. Show the Wait for Clap block and the Sound sensor and create an example program together. Run the program, and have students discuss what the robot is doing. Introduce the term event, which is an action that causes something to happen. The action here is the clap, which causes KIBO to continue its program. All of the sensors that KIBO has (sound, light, and distance) use events to trigger KIBO which they will experiment with in later lessons.

What is the Sound Sensor?

KIBO's **Sound Sensor** is shaped like an ear and senses sounds from the environment. It is programmed using the Wait for Clap block. In the example program, KIBO will turn right, wait for a loud sound (like a clap) before it spins and ends.

SHAPE SHIFTING (15 min)

Show students the large KIBO Says cards depicting the three orange Sound Recorder blocks: Play Circle, Play Square, and Play Triangle. First demonstrate the game as a class. Ask for three student volunteers to choose a sound or action to go with each shape. For example, a student might decide to jump on one foot for the Circle, another student might decide to yell "Hooray!" for the Triangle, and a third student might decide to hold up a book for the Square. To play the game, explain to students that when you hold up one of the three shape cards, they should only perform the action associated with that shape.

Once students are comfortable with the game, split into small groups and have students take turns deciding the actions for the shapes. Have students recreate the shape cards using construction paper and markers. The purpose of this activity to get students comfortable with cause and effect; students must shift or alter their actions depending on the shape of the card. This will also help students better understand how the KIBO Sound Recorder module works in the next demonstration activity.

What is the Sound Recorder?

KIBO's sound recorder/playback module has three different buttons – square, triangle, and circle– that allows students to record three different short sound clips. Remember to match the shape of the block to the recorded sound!

KIBO SOUND RECORDER (5 min)

Show students the KIBO Sound Recorder module. Demonstrate with a model program how they can make three different recordings by pressing and holding down on the three shape buttons on the module. Note that the Sound Recorder must be connected to power by inserting the module into the KIBO body before recording.

FREE PLAY (15 min)

Individually or in pairs, students should take this time to explore the Sound Sensor and Sound Recorder modules freely. By the end of this free-exploration, students should understanding the difference between sound input (i.e., KIBO needs to hear the clap using the Sound Sensor before proceeding) and sound output (i.e., students record the specific sounds that they want KIBO to play using the Sound Recorder). Encourage students to try other noises, like stomping or ringing a bell, to trigger the Sound Sensor!

SOLVE-IT ASSESSMENT B (10 min)

On the Appendix B-2 you will find assessment B. Please hand out one assessment sheet to every child in your class.

Instructions:

- Read each question and option out loud to the group. Students can ask to have questions or options read out loud up to 3 times.
- Instruct children to circle only 1 answer per question.
- Make sure students answer the questions by themselves. Students should not be discussing or copying answers.
- Hand in completed answer sheets to Angela de Mik or a member of your school's assessment team.

Lesson 7: Cause and Effect - Level 2

Powerful Idea From Computer Science:
Algorithms, Modularity, Representation

Powerful Idea From Literacy:
Descriptive Language in Writing

OVERVIEW

Students will engage in goal-oriented programming using the Sound Sensor and Sound Recorder modules. In this lesson, students combine their programming knowledge from previous lessons to program their robots to sing and dance to the “If You’re Happy and You Know It” song.

PURPOSE

This lesson reinforces students’ learning of the KIBO blocks and the use of different KIBO modules. At this point in the curriculum, students should be familiar with the Lightbulb, Sound Sensor, and Sound Recorder modules. The focus of the project reflection time will shift from discussing debugging issues (though that, of course, leads to thoughtful discussion) to goal-oriented programming and how students’ initial ideas and plans might not always translate to their final KIBO program.

ACTIVITIES

- Sing “If You’re Wild and You Know It” (5 min)
- Program "If You're Wild and You Know It" (30 min)
- Project Reflection (10 min)
- Share Creations (15 min)

STUDENTS WILL BE ABLE TO...

- Program KIBO to sing and dance to the “If You’re Wild and You Know It” song

PREPARATION FOR TEACHERS

- ☐ Read through the Activity Guide
- ☐ Ensure all KIBO bodies have 4 working AA batteries
- ☐ Sort KIBO blocks and pieces (listed in Materials section) by part and place in a central location

MATERIALS

FOR THE TEACHER:

- 1 flathead screwdriver
- Extra AA batteries
- Discussion Sentence Starters anchor chart*

FOR STUDENTS:

- Design Journal (see Appendix C for example)
- KIBO bodies, wheels, motors, Begin and End blocks, blue Motion blocks, yellow Light blocks, Beep and Sing blocks, Wait for Clap blocks, Sound Sensors, Sound Recorder module, orange Sound Recorder blocks
- KIBO stickers

*See Appendix A for example

VOCABULARY

- Senses — the way humans and animals take in information about the surrounding environment. Humans have five senses: touch, taste, smell, sight, and hearing
- Sensor — a special part that helps machines take in information about the surrounding environment; there are sensors that are very much like human senses
- Event — an action that causes something to happen
- Circle – a round, closed shape with no edges
- Record — to make something (like a sound) permanent so that it can be played back at a later time
- Sound — a type of energy made by vibrations in the air that we can hear
- Square — a closed shape with four equal sides
- Triangle — a closed shape with three sides

Lesson 7: Activities

IF YOU'RE WILD AND YOU KNOW IT (5 min)

In the story *Where the Wild Things Are*, Max acts wild. Students will work individually or in pairs to program KIBO to dance a version of the song "If You're Happy and You Know It," Wild-fied, "If You're Wild and You Know It". Sing the song together as a class.

If you're wild and you know it, clap your hands (clap-clap)
If you're wild and you know it, clap your hands (clap-clap)
If you're wild and you know it, then your face will surely show it
If you're wild and you know it, clap your hands. (clap-clap)

2. stomp your feet
3. shout "Hooray!"
4. do all three (clap-clap, stomp-stomp, "Hoo-ray!")

PROGRAM "IF YOU'RE WILD AND YOU KNOW IT" (30 min)

Take out KIBOs and blocks. Before students begin programming KIBO to dance, have students take out their Design Journals and write a response to the following prompt: *What do you wish your robot would do to show that she/he is happy (e.g., dance a particular dance, sing a particular song)*? Have a few students read out their responses and then explain that today KIBO will follow the instructions of the classic song "If You're Happy and You Know It," but in a few lessons, the students will get to decide for themselves what KIBO does to show she/he's happy!

Students should program their robots to move in any wild way during the lyrics "If You're Wild and You Know It" but include the program instructions that have KIBO wait to hear a clap (representing the lyrics "Clap your Hands") before KIBO begins moving. Students can choose as few or as many blocks as they would like to put after the "Wait for Clap" block.

This is an example of a program that teachers can use as a model:

If You're Wild and You Know It – Sample Program

PROJECT REFLECTION (10 min)

Before sharing their projects, have students take out their Design Journals and use the KIBO stickers to write out their program. Where did they choose to place the Wait for Clap block? What was fun or challenging about creating their program? Did their program get KIBO to do what they wanted? Students should document their reflections in their Design Journals and are encouraged to bring their reflections to share in the Technology Circle.

SHARE CREATIONS (15 min)

Have students sit in a technology circle to share their programs. Encourage students to verbalize their thinking and reasoning behind their program. For example, *ask students: Where did you decide to add the Wait for Clap block? What were the different ways you tried to trigger the Sound sensor (clapping, talking, etc.)? What kinds of sounds did you record? Why did you choose a particular block in your program?* Students can also use the discussion sentence starters from the anchor chart to talk about their creations.

Lesson 8: Repeat Loops - Level 1

Powerful Idea From Computer Science:
Control Structure, Modularity

Powerful Idea From Literacy:
Repetition as a Literacy Device, Repetition in Word Forms

OVERVIEW

In this lesson, students understand the importance of repetition both in computer science and literature. Students will learn about a new instruction that makes KIBO repeat programming instructions infinitely or a given number of times. Students also think about repetition as a literary device and the purpose it serves in a text, as well as repetition in word structure as a review of foundational phonic and word recognition skills.

PURPOSE

The activities in this lesson broaden students' understanding of patterns by highlighting the different ways that repetition can be used to make something more efficient or more entertaining. Students also begin to learn that there are multiple ways of representing the same outcome, and that repeat loops are one way that computer scientists make more efficient programs.

ACTIVITIES

- Repetition in Stories and Songs (15 min)
- Toothbrush Exercise (15 min)
- KIBO Repeat with Numbers (20 min)
- Solve-It Assessment C (10 min)

STUDENTS WILL BE ABLE TO...

- Identify patterns in code sequences and rewrite codes using repeat loops
- Use KIBO number parameters to make a program that loops a certain of times
- Understand how repetition is used in stories and songs

PREPARATION FOR TEACHERS

- ☐ Read through the Activity Guide
- ☐ Ensure all KIBO bodies have 4 working AA batteries
- ☐ Sort KIBO blocks and pieces (listed in Materials section) by part and place in a central location
- ☐ Print Solve-It Assessment C (one for each student)

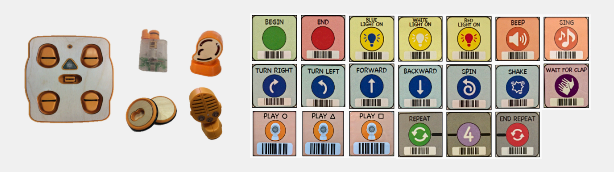
MATERIALS

FOR THE TEACHER:

- KIBO Says cards
- 1 flathead screwdriver
- Extra AA batteries

FOR STUDENTS:

- Design Journal (see Appendix C for example)
- Small KIBO block cutouts (one set for each pair of students)
- KIBO bodies, wheels, motors, Begin and End blocks, blue Motion blocks, yellow Light blocks, Beep and Sing blocks, Wait for Clap blocks, Sound sensors, Sound Recorder module, orange Sound Recorder blocks, Repeat and End Repeat blocks, number parameters



VOCABULARY

- Loop – something that repeats over and over again
- Parameter – a value or limit given to a robot that can be changed (e.g., programmer sets the limit for how many times a robot repeats a sequence)
- Pattern – a design or sequence that repeats
- Repeat – to do something more than once

Lesson 8: Activities

REPETITION IN STORIES AND SONGS (15 min)

Throughout the book *Where the Wild Things Are*, certain phrases and words are repeated multiple times. Reread the "terrible pages" and come up with a list of what phrases or words are repeated (e.g., terrible, wild things, eat you up, etc.). *Ask students: Why might the author have done that? What purpose does that serve for the reader?* The purpose of this activity is to remind students that repetition is essential in language, literature, and, as they will learn today, coding as well.

OR

If you feel the students/class could use a break from *Where the Wild Things Are*, choose a song the students like instead. Hand out the lyrics to the class, play the song for the class, and ask students, as the song is playing, to circle repeating stanzas. The purpose of this activity is to remind students that repetition is essential in language, literature, and, as they will learn today, coding as well.

TOOTHBRUSH EXERCISE (15 min)

Have students think about the way they brush their teeth. *Ask students: Are there actions that you have to repeat? (e.g. moving the toothbrush from left to right) Are there motions that only happen once? (e.g. squeezing out toothpaste)* Working in pairs, have students write a program in their Design Journals for brushing their teeth. Have students act it out to ensure they have covered all the steps.

Once pairs finish, have several students share out their programs. As a class, discuss how the programs were similar or different.

KIBO REPEAT WITH NUMBERS (20 min)

Take out the KIBOs and blocks. Using the large KIBO Says cards first, show students a sample KIBO program that has repeating blocks (see examples below). *Ask students: What is the repeating pattern in this program? How many times does it repeat?*

Identifying Patterns

In the program below, the repeating pattern is [White Light, Beep] and occurs four times.

In the program below, the repeating pattern is [Spin, Wait for Clap, Sing] and occurs twice. Note that the White Light block is not part of the repeating pattern.

As a class, look back at your example KIBO programs with repeating patterns. *Ask students: Is there a way I could make this program shorter?* Demonstrate to students that the Repeat and End Repeat blocks can be used to make programs that are shorter and more efficient.

Make a sample program using the Repeat blocks and the Repeat Forever parameter card. Emphasize that the robot only repeats the instructions in between the Repeat and the End Repeat blocks. Note to students how the robot will not stop unless you press the button (to stop it). Try another model program using the Repeat 2, 3, or 4 parameters.

Distinguish this kind of repetition from literature, where a repetition may take place pages apart and can include slight variations. For example, in the story *Where the Wild Things Are*, the word "terrible" was repeated several times in different parts of the story.

What is a Repeat Loop?

Repeat and End Repeat are like the bread of a sandwich. The programming blocks put inside of them are like the filling. KIBO will only repeat commands that are placed inside of the **Repeat Loop** sandwich. Segments of the code placed outside of the sandwich will not be repeated.

Parameters are used to tell the robot how many times to repeat, or when to stop repeating. In the program below, KIBO will shake 4 times.

Have students explore their own programs using the Repeat blocks. The emphasis here should be using proper syntax, rather than scanning the program onto KIBO. One suggestion for this activity is to have students create their KIBO programs using the blocks first. Then, students can move to a testing station in a designated location in the classroom, where they can test to make sure their programs work.

SOLVE-IT ASSESSMENT C (10 min)

On the Appendix B-3 you will find assessment C. Please hand out one assessment sheet to every child in your class.
Instructions:

- Read each question and option out loud to the group. Students can ask to have questions or options read out loud up to 3 times.
- Instruct children to circle only 1 answer per question.
- Make sure students answer the questions by themselves. Students should not be discussing or copying answers.
- Hand in completed answer sheets to Angela de Mik or a member of your school's assessment team.

Lesson 9: Repeat Loops - Level 2

Powerful Idea From Computer Science:
Control Structures, Debugging, Sensors

Powerful Idea From Literacy:
Descriptive Language, Perspectives in Narrative

OVERVIEW

In this lesson, students will learn about the Light and Distance sensors and build upon their understanding of Repeat Loops. Students will program their robots to perform different actions using the Repeat blocks with number and sensor parameters. Based on students' level of understanding, educators should feel free to introduce only one of the sensors. Light and Distance sensors can be tricky, as it can be challenging to create the appropriate environment to trigger the sensors.

PURPOSE

Robots use sensors to gather information from their environment. Students will experiment with the Light and Distance sensors and continue to reflect upon human senses and how they differ from robot sensors. Free exploration of the sensors will allow students to test the sensitivity of the sensors by setting up different types of environments.

ACTIVITIES

- My Five Senses (20 min)
- KIBO Repeat with Sensors (15 min)
- Free Play with Repeats (25 min)

STUDENTS WILL BE ABLE TO...

- Compare and contrast human senses and robot sensors
- Successfully test a KIBO program using the Light and Distance sensors

PREPARATION FOR TEACHERS

- ☐ Read through the Activity Guide
- ☐ Print Design Journals (one for each student)*
- ☐ Ensure all KIBO bodies have 4 working AA batteries
- ☐ Sort KIBO blocks and pieces (listed in Materials section) by part and place in a central location

MATERIALS

FOR THE TEACHER:

- 1 copy of *Where the Wild Things Are* by Maurice Sendak
- 1 flathead screwdriver
- Extra AA batteries

FOR STUDENTS:

- Design Journal (see Appendix C for example)
- KIBO bodies, wheels, motors, Begin and End blocks, blue Motion blocks, yellow Light blocks, Beep and Sing blocks, Wait for Clap blocks, Sound sensors, Sound Recorder module, orange Sound Recorder blocks, Repeat and End Repeat blocks, number parameters, Light and Distance sensors, gray sensor parameters

*See Appendix A for examples

VOCABULARY

- Distance - the amount of space between two things or people
- Senses – the way humans and animals take in information about the surrounding environment. Humans have five senses: touch, taste, smell, sight, and hearing
- Sensor – a special part that helps machines take in information about the surrounding environment; there are sensors that are very much like human senses
- Event – an action that causes something to happen

Lesson 9: Activities

MY FIVE SENSES (20 min)

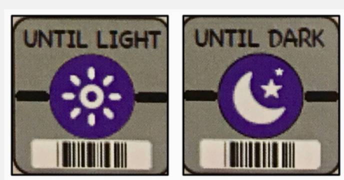
Pass out students' Design Journals. Read the story *My Five Senses* by Aliki and discuss scenarios in which students might use each of their five senses. Then, in their Design Journals, have students choose one of the five senses and draw a picture of a situation in which they would use that sense.

KIBO REPEAT WITH LIGHT SENSOR (15 min)

Take out the KIBOs and blocks. Explain to students that KIBO has some of the same senses that we (the ability to "see" light and dark, to feel touch, to speak) and that today they will learn how KIBO is able to "see" light and dark). Explain that KIBO needs special programming instructions to tell KIBO what to do with the information from its Light sensor. Show the Repeat and End Repeat blocks, which are now familiar, and the new Until Light/Until Dark parameter cards. Create two example programs together, one which uses the Until Light parameter and the other with the Until Dark parameter. Run the programs, and have students discuss what the robot is doing in each scenario.

What is the Light Sensor?

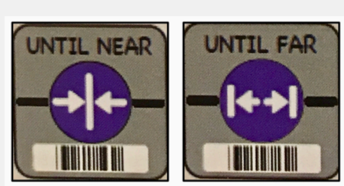
KIBO's **Light Sensor** can detect light in the room around it. If a flashlight is shining on KIBO, the light sensor will tell KIBO that it is bright. If there are no lights shining on KIBO, the light sensor will tell KIBO that it is dark.



Then introduce the Distance sensor. Create two example programs together, one which uses the Until Near parameter and the other which uses the Until Far parameter. Run the programs, and have students discuss what the robot is doing.

What is the Distance Sensor?

KIBO uses the **Distance Sensor** to see how near or far KIBO is from other objects. With Distance Parameters, the Distance Sensor can be used with Repeat Loops to control how KIBO moves.



FREE PLAY WITH REPEATS (25 min)

Individually or in pairs, students will create programs using the Light and Distance sensors. Free play with the sensors will allow students to tinker and explore the sensitivity of the sensors. Students can shine a flashlight to trigger the Light sensor or place objects in front of the robot, triggering the Distance sensor. Emphasize that the Repeat blocks with sensor parameters mean that KIBO will continue to perform the actions inside of the Repeat loop until the environment changes to the specific parameter.

Lesson 10: If Statements

Powerful Idea From Computer Science:
Control Structures, Debugging, Sensors

Powerful Idea From Literacy:
Identify Conflict and Resolution Making Predictions

OVERVIEW

In this lesson, students build upon their explorations of the Light and Distance sensors but this time with conditional statements, in which the sensor checks for the state of the environment only once before the robot decides which action to perform. Students will program their robots to take different actions based on the state of the sensor.

PURPOSE

This lesson allows students to connect their understanding of branched programs with the key literacy concepts of cause and effect and making predictions. Knowing that outcomes can vary depending on the circumstances is an important concept in early childhood, as students begin to comprehend how decisions are made in everyday life.

ACTIVITIES

- Writing an Alternative Story (20 min)
- KIBO If Statements (20 min)
- Free Play with Conditionals (20 min)

STUDENTS WILL BE ABLE TO...

- Successfully test a conditional KIBO program using the Distance and Light sensors
- Identify situations that would require an If statement or a Repeat loop

PREPARATION FOR TEACHERS

- ☐ Read through the Activity Guide
- ☐ Ensure all KIBO bodies have 4 working AA batteries
- ☐ Sort KIBO blocks and pieces (listed in Materials section) by part and place in a central location

MATERIALS

FOR THE TEACHER:

- 1 copy of *Where the Wild Things Are* by Maurice Sendak
- Optional: 1 copy of *Hand, Hand, Fingers, Thumb* by Al Perkins
- 1 flathead screwdriver
- Extra AA batteries

FOR STUDENTS:

- Design Journal (see Appendix C for example)
- Flashlights (one per pair of students)
- KIBO bodies, wheels, motors, Begin and End blocks, blue Motion blocks, yellow Light blocks, Beep and Sing blocks, Wait for Clap blocks, Sound sensors, Sound Recorder module, orange Sound Recorder blocks, Repeat and End Repeat blocks, number parameters, Light and Distance sensors, gray sensor parameters, If and End If blocks, purple sensor parameters

VOCABULARY

- Branched program — a program with two or more possible sequences; the computer/robot makes its decision based on an event
- Conditional — only happens sometimes
- Event — an action that causes something to happen

Lesson 10: Activities

WRITING AN ALTERNATIVE STORY (20 min)

The purpose of this activity to have students think creatively about what could have happened in *Where the Wild Things Are* if Max had done things differently. If necessary, reread the story to students.

Below are some examples from the story. Ask students to think about these hypothetical scenarios, and have several students share out their hypotheses.

- *What would have happened if Max hadn't felt wild and yelled?*
- *If Max had made mischief of one kind, but not another…*
- *If Max had not responded to his mother, "I'll eat you up,"…*
- *If Max had been afraid of the wild things, then…*
- *If Max had never stopped the wild rumpus, then…*

Now the students have the opportunity to turn these suggested alternative stories into compositional texts.

Students will write their alternative stories in their Design Journals. This activity is also an opportunity to review whatever skills the students have most recently learned in writing (e.g. strategies for organization, capitalization of proper nouns, etc.).

KIBO IF STATEMENTS (20 min)

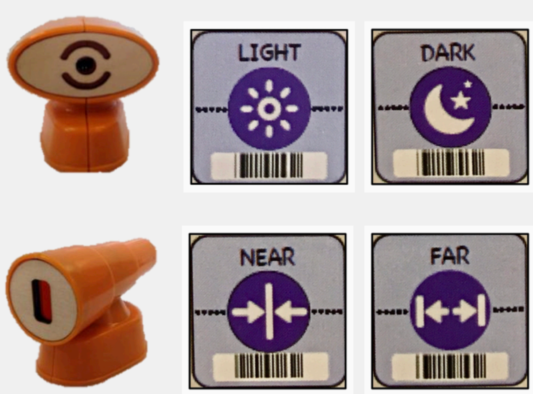
Explain to students that in the programs they have learned so far, KIBO has only one choice of what instructions to do next. Now they will learn an instruction that gives KIBO two choices, and the Light and Distance sensors will help KIBO decide which set of instructions to follow each time the program is run.

Introduce the If and End If blocks, as well as the Near/Far and Light/Dark parameters. Demonstrate what happens when you do and do not put your hand in front of the Distance sensor when the Near/Far parameters are used. Create another program using the Light sensor and Light/Dark parameters and demonstrate what happens when you do and do not shine a light into the sensor.

What is an If Statement?

If Statements allow KIBO to make a choice based on what it can sense, just like your students can! Use these four parameters with If Statements. Remember to attach the appropriate sensors!

Note that unlike Repeat Loops where KIBO will keep repeating the commands inside the loop until it senses something, KIBO's sensor will only check one time before moving to the next part of the program.



Discuss situations in the real world where someone may have to make a choice depending on the circumstances of the environment. For example, “If it is rainy out, I will bring an umbrella”. Connect this idea to the Writing an Alternative Story activity, in which students came up with creative endings to the *Where the Wild Things Are* story.

FREE PLAY WITH CONDITIONALS (20 min)

Take out the KIBOs and blocks. Let students explore building programs with the If and End If blocks. This exploration gives them a chance to learn how to use the blocks in a program, think of situations that require it, and further understand how to use sensors.

Students sometimes have the misconception between Repeat loops with sensors and If statements with sensors. This is important to identify and clarify with demonstrations. *Ask students: What would happen if I replaced your If statement with a Repeat loop instead? Would I need to change the parameter card? (Yes) How does that change the outcome of the program?*

EXTENDED ACTIVITIES:
HAND, HAND, FINGERS, THUMB

If your students need more time to explore the idea of repetition, read the book Hand, Hand, Fingers, Thumb by Al Perkins to students. Re-read the book, this time asking students to raise their hand (or make another movement) every time they hear repetition. Pause reading, and write the repeated word or phrase on the board.

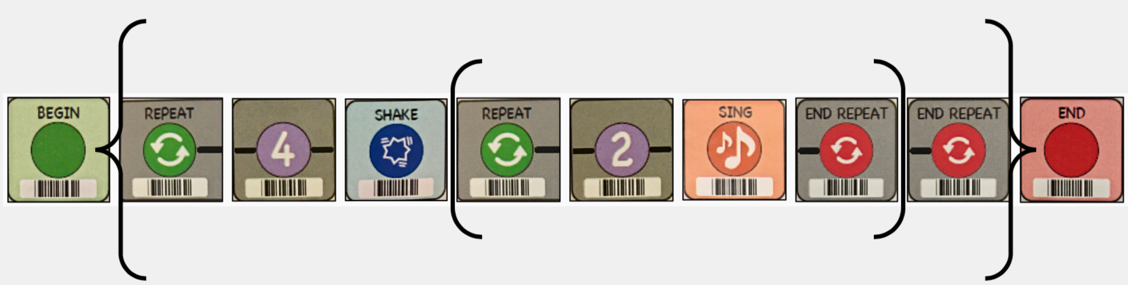
At the end of the read-aloud, lead a student-centered discussion about the different kinds of repetition in the book. Explain to students how some of the words repeated within a single page, whereas others were repeated across pages. This is an example of how writers use repetition in different ways. Make the connection that KIBO can also use repetition in different ways. KIBO can have repeats within other repeats, ifs within other ifs - these are called **nested loops** and **nested statements**.

NESTED STATEMENTS

If time permits, show examples of nested loops and nested statements. These are the most complex kinds of programs students can create using KIBO. **Nested loops** are a way to make even more efficient programs, in which parts of instructions are repeated a different number of times. Show an example of a nested loop to students. *Ask students: Which blocks are in the **inner** loop? How many times does it repeat? Which blocks are in the **outer** loop? How many times does it repeat?*

What is a Nested Loop?

A **nested loop** is a loop inside of another loop. For instance, in the program below, KIBO will shake and beep twice, and repeat this sequence four times.



Just like there can be a repeat inside of another repeat, nested statements are created using any kind of statement (if or repeat) inside of another statement (if or repeat). Before showing the example of a nested statement with KIBO, have students imagine how an automatic faucet works. *Ask students: What happens when you put your hand close to the sensor? Does this happen every time you make that movement?* Walk through the scenario with students: Every time (repeat forever) you move your hands close to the sensor (if), the water runs (end if) (end repeat).

Demonstrate another example with a stoplight. *Ask students: What do you do if the light turns green? If the light turns yellow? If the light turns red?* Explain to students how they could use multiple if statements to demonstrate how a driver would make a decision depending on the color of the stoplight.

What is a Nested Statement?

A **nested statement** is any statement inside of another statement. This could mean a repeat loop inside of an if statement, an if statement inside of a repeat loop, an if statement inside another if statement, etc. You can have as many nested statements in a program as you'd like, but make sure that each statement has its appropriate start and end block! For example, when the program below is running, KIBO will shake every time it senses that it is near something.

Lesson 11: Final Project - Writing the Wild Rumpus Composition

Powerful Idea From Computer Science:	**Powerful Idea From Literacy:**
Design Process	Writing Process

OVERVIEW

Students will begin their final project in this lesson: writing their Wild Rumpus composition. During the course of this final project, students will put to use all the concepts learned during the previous lessons. Students can use parts of their programs from previous lessons, but they should be encouraged to start fresh and transfer their skills to a new context.

PURPOSE

The purpose of the final project is to allow students to demonstrate the skills they have acquired throughout the previous lessons and to apply them in new, creative ways. By writing out their plans first before building with KIBO, students make purposeful decisions about their projects and understand that not all ideas on paper can transfer to the actual design.

ACTIVITIES

- Wild Rumpus Composition (30 min)
- Writing vs. Coding (5 min)
- Peer Feedback (10 min)
- Collaboration Web (5 min)
- Begin Coding the Wild Rumpus (20 min)

STUDENTS WILL BE ABLE TO...

- Utilize the Writing Process by writing their Wild Rumpus composition
- Decide which of their ideas can and cannot be translated into KIBO programs
- Identify and show appreciation to those who have helped them with their final projects

PREPARATION FOR TEACHERS

- ☐ Read through the Activity Guide
- ☐ Print Collaboration Webs (one for each student)
- ☐ Ensure all KIBO bodies have 4 working AA batteries
- ☐ Sort KIBO blocks and pieces (listed in Materials section) by part and place in a central location

MATERIALS

FOR THE TEACHER:

- 1 flathead screwdriver
- Extra AA batteries

FOR STUDENTS:

- Design Journal (see Appendix C for example)
- Index cards (one for each student)
- KIBO bodies, wheels, motors, Begin and End blocks, blue Motion blocks, yellow Light blocks, Beep and Sing blocks, Wait for Clap blocks, Sound sensors, Sound Recorder module, orange Sound Recorder blocks, Repeat and End Repeat blocks, number parameters, Light and Distance sensors, gray sensor parameters, If and End if blocks, purple sensor parameters, platform pieces

Lesson 11: Activities

WILD RUMPUS COMPOSITION (30 min)

First ask students to take an index card and write down their top three activities they would have in their own Wild Rumpus. *You might first provide an example: In my Wild Rumpus, I would have an awesome dance party, some howling at the moon, and making s'mores. Ask students: What three things would you have in your Wild Rumpus?* Students should refer back to this index card throughout the final project.

Students will engage in the Writing Process to plan out their Wild Rumpus composition using their Design Journals. Based on their three ideas from the index card, explain to students they will write a story about their Wild Rumpus. Below are examples of things to include in their composition, as well as writing tips:

- Identify the audience and purpose of writing *(Who will be reading your ideas for your Wild Rumpus? What might they want to know about your project?)*
- Use prewriting strategies to generate ideas before writing Use organizational strategies to keep track of the different project components
- Organize writing to include a beginning, middle, and end *(How does the Wild Rumpus start, what happens during it, and how does it end?)*

Organize writing to include a beginning, middle, and end (How does the Wild Rumpus start, what happens during it, and how does it end?)

WRITING VS. CODING (5 min)

This activity provides a chance for students to reflect on the constraints and affordances of each medium, writing and coding. Have students come together in a technology circle. *Ask students:*

- *Are there certain activities you wrote about that you can code with KIBO?*
- *Are there certain activities you wrote about that might not work with KIBO? How will you change your idea so that it makes sense for KIBO?*

COLLABORATION WEB (5 min)

Pass out each student's Collaboration Web. Have students draw lines on their Collaboration Webs from their picture in the middle to pictures of other students in the class who gave them good pieces of feedback or helped in a different way. Encourage students to continue filling out their Webs as they begin to create their final projects with KIBO.

BEGIN CODING THE WILD RUMPUS (20 min)

Take out the KIBOs and blocks. Using their design plans, students will plan how to turn their compositional Wild Rumpus into a programmed Wild Rumpus. Encourage students to challenge themselves and make program plans that will use advanced parts such as sensors, Repeat loops, and If statements. Encourage students to keep track of their progress using their Design Journals.

Lesson 12: Final Project - Coding the Wild Rumpus

Powerful Idea From Computer Science:
Design Process

Powerful Idea From Literacy:
Writing Process

OVERVIEW

In this final lesson, students will code their Wild Rumpus programs. During the course of this final project, students will put to use all the concepts learned during the previous lessons. When students are finished with their projects, they will share them with each other and offer their gratitude to those who have helped them along the way.

PURPOSE

The purpose of the final project is to allow students to demonstrate the skills they have acquired throughout the previous lessons and to apply them in new, creative ways. After students finish presenting their projects to their peers, educators are encouraged to invite families and community members to view students' final projects.

ACTIVITIES

- Coding the Wild Rumpus (20 min)
- Share Creations and Deliver Cards (15 min)
- Wild Rumpus Reflection (10 min)
- Solve-It Assessment D (15 min)

STUDENTS WILL BE ABLE TO...

- Demonstrate the Design Process in full by planning, designing, and creating a final KIBO project
- Share final projects with peers, family and community members
- Identify and show appreciation to those who have helped them with their final projects

PREPARATION FOR TEACHERS

- ☐ Read through the Activity Guide
- ☐ Rent the movie *Where the Wild Things Are* from the local library or movie rental kiosk (optional)
- ☐ Ensure all KIBO bodies have 4 working AA batteries
- ☐ Sort KIBO blocks and pieces (listed in Materials section) by part and place in a central location
- ☐ Print Solve-It Assessment D (one for each student)

MATERIALS

FOR THE TEACHER:

- 1 copy of *Where the Wild Things Are* by Maurice Sendak
- *Where the Wild Things Are* movie (optional)
- 1 flathead screwdriver
- Extra AA batteries

FOR STUDENTS:

- Design Journal (see Appendix C for example)
- Crafts and recycled materials
- Construction paper or other kind of paper to write thank you letters
- KIBO bodies, wheels, motors, Begin and End blocks, blue Motion blocks, yellow Light blocks, Beep and Sing blocks, Wait for Clap blocks, Sound sensors, Sound Recorder module, orange Sound Recorder blocks, Repeat and End Repeat blocks, number parameters, Light and Distance sensors, gray sensor parameters, If and End if blocks, purple sensor parameters, platform pieces

Lesson 12: Activities

CODING THE WILD RUMPUS (20 min)

Take out KIBOs and blocks. Students will continue working on coding their final Wild Rumpus projects. As a class, create a backdrop for the Wild Rumpus using butcher paper and other crafts and recycled materials. Students can create one big mural together or create individual scenery for their robots.

As students work on their final projects, they should also be filling out their Collaboration Web. *Ask students: Count all the lines you have drawn between yourself and each of the students in the class. Write the number next to each student's picture on your web. For which students did you draw the most lines?*

Using construction paper or other kind of nice paper, have students write three thank you cards to the three students who have helped them the most.

SHARE CREATIONS AND DELIVER CARDS (15 min)

During the final presentations, have students present their Wild Rumpus compositions and KIBO projects. Students can share their final projects altogether in a technology circle, or as a gallery walk, in which half of the students walk around the classroom to each project while the other half present their projects. Then the two groups switch. Students should share:

- their robots and decorations
- why they chose the features they did for their robot
- the final program and what each block and module represent
- anything that was hard, easy, surprising, interesting, etc. about the process.

Take photos of students' final projects and KIBO codes.

WILD RUMPUS REFLECTION (10 min)

If more time is needed for students to finish their final projects, this reflection activity can be assigned for homework. Now that students have written a Wild Rumpus composition and created a Wild Rumpus project with KIBO, have students write a letter to their families explaining their projects. *Ask students: What was your project about? What did you learn by playing with KIBO? What was your favorite thing? What was your most challenging thing?*

Send students' letters to families, along with pictures of their compositions, final projects and KIBO codes.

SOLVE-IT ASSESSMENT D (15 min)

On the Appendix B-4 you will find assessment D. Please hand out one assessment sheet to every child in your class.
Instructions:

- Read each question and option out loud to the group. Students can ask to have questions or options read out loud up to 3 times.
- Instruct children to circle only 1 answer per question.
- Make sure students answer the questions by themselves. Students should not be discussing or copying answers.
- Hand in completed answer sheets to Angela de Mik or a member of your school's assessment team.

Appendix A. Materials

Appendix A. Materials

Robotics and Technology Materials:

- ❒ KIBO 21 robotics kits: It is recommended to deconstruct the kits into individual parts or to provide each pair of students with one pre-made kit, so that students can access only the parts required for each lesson.
- ❒ 1 flathead screwdriver
- ❒ AA batteries (each KIBO kit requires 4 AA batteries)
- ❒ Extra AA batteries
- ❒ Flashlights (one per pair of students)

Art and Game Materials:

- ❒ Construction paper or other kind of decorative paper
- ❒ Crafts and recycled materials (e.g. scrap paper, scissors, straws, popsicle sticks, recycled cardboard, any other available materials that students can use to decorate their robots)
- ❒ Masking tape
- ❒ Index cards (one for each student)

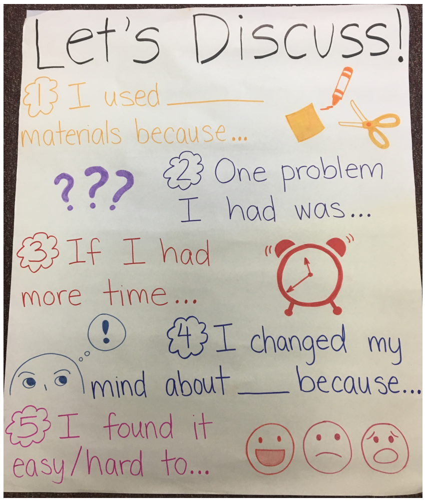
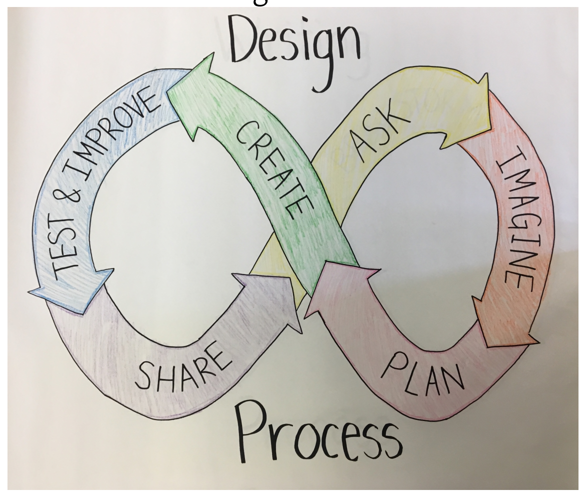
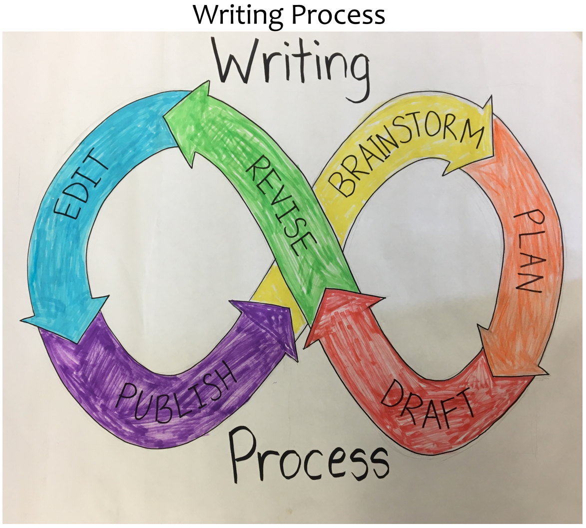
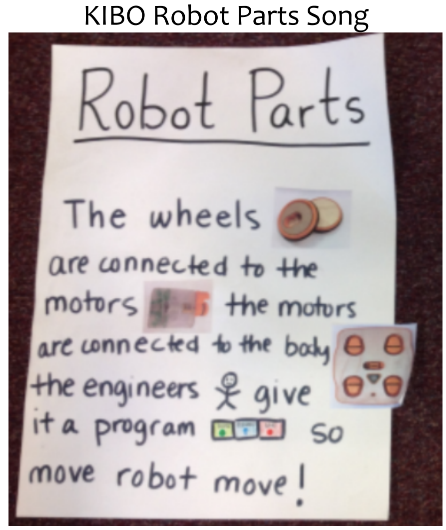
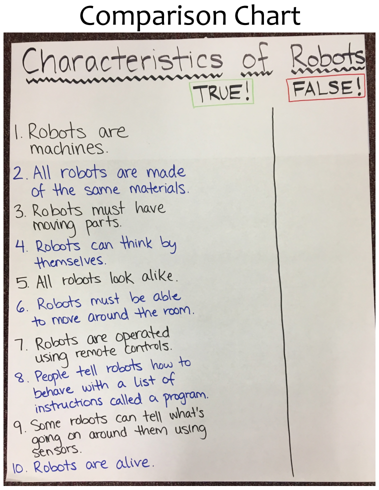
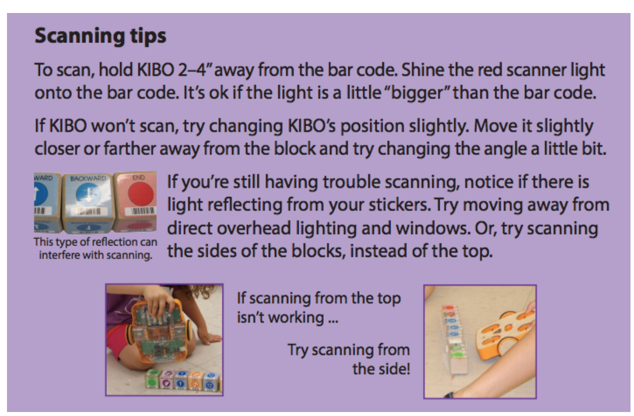
Teaching Materials:

- ❒ 1 copy of *Where the Wild Things Are* by Maurice Sendak (it might be helpful to have multiple copies for students to reference during projects)
- ❒ 1 copy of *My Five Senses* by Aliki
- ❒ 1 copy of *Hand, Hand, Fingers, Thumb* by Al Perkins
- ❒ *Where the Wild Things Are* movie (optional)
- ❒ Large KIBO Says cards (purchase from KinderLab Robotics or make your own)
- ❒ Premade anchor charts **(see following pages for examples)**
  - ❒ Discussion sentence starters
  - ❒ Design Process
  - ❒ Writing Process
  - ❒ KIBO Robot Parts song
  - ❒ Characteristics of Robots comparison chart
  - ❒ Why is KIBO Confused?
- ❒ Printed pictures **(see following pages for examples)**
  - ❒ 8-10 pictures of naturally occurring and man-made objects
  - ❒ 8-10 pictures of robots and non-robots
  - ❒ Large letter cards: A, R, C
  - ❒ Handwritten and typed message

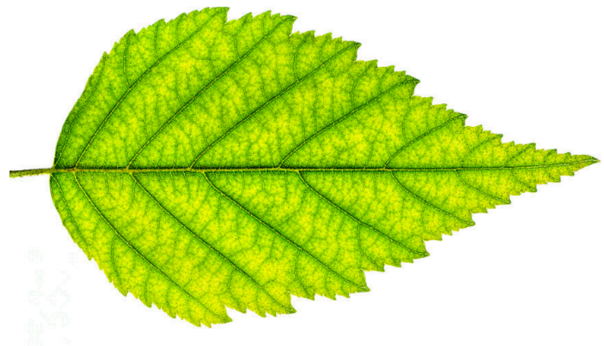
Optional Materials:

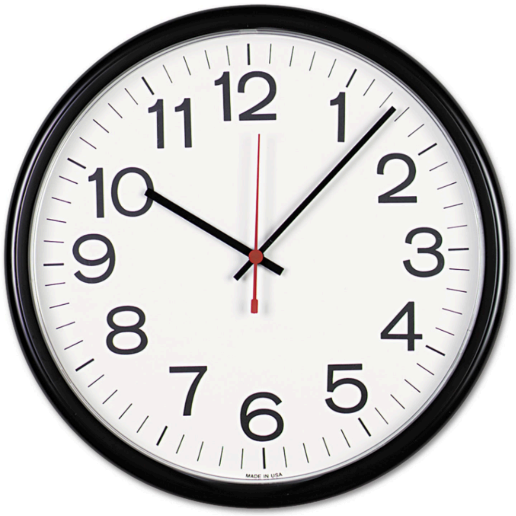
- ❒ Small KIBO block cutouts (download from http://bit.ly/KIBOcutouts)
- ❒ KIBO Parts Bingo cards (download from http://bit.ly/KIBOpartBINGO)
- ❒ KIBO Blocks Bingo cards (download from http://bit.ly/KIBOblockBINGO)

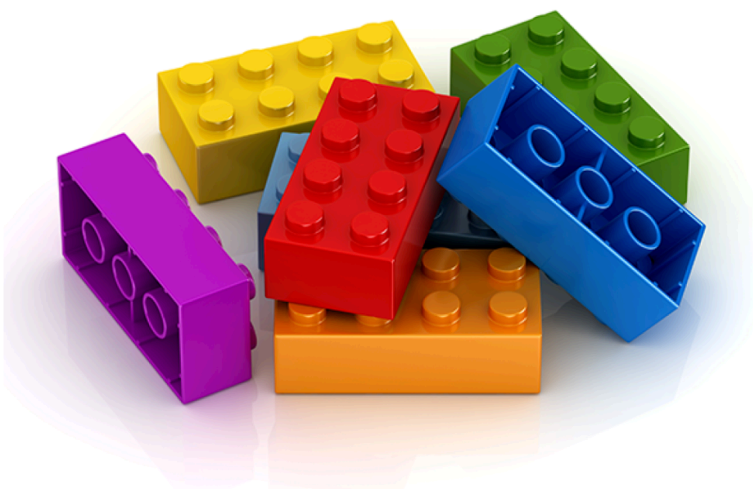
Examples of Anchor Charts:

Discussion Sentence Starters 	Design Process 
Writing Process 	KIBO Robot Parts Song 
Characteristics of Robots Comparison Chart 	Why is KIBO Confused? 

Examples of Printed Pictures:
8-10 pictures of naturally occurring and man-made objects







Examples of Printed Pictures:

8-10 pictures of robots and non-robots

For non-robot pictures: reuse the images from the man-made objects in Lesson 1

Examples of Printed Pictures:
Large letter cards: A, R, C

A

R

C

Handwritten and typed message

Molly went to the store to buy apples and crackers.

Molly went to the store to buy apples and crackers.

Appendix B-1. Solve-It Assessment A

Name ______________________________ Date __________________

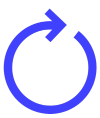
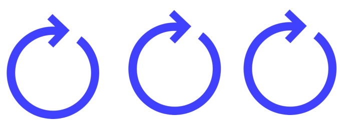
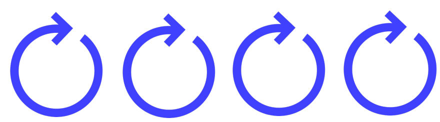
Teacher ___________________________

Circle the correct answers.

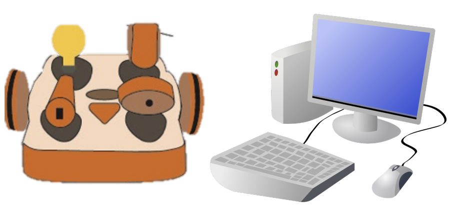
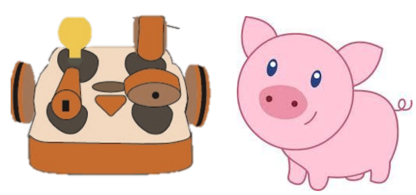
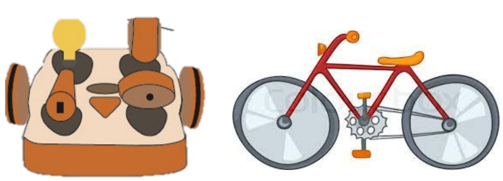
1. Which of these things would you need a design process for?

Keeping things the same	Making new things	Breaking something
		

2. How many times can you go through the writing process and design process?

1 time	3 times	You can repeat the cycle as many times as you need
		 . . .

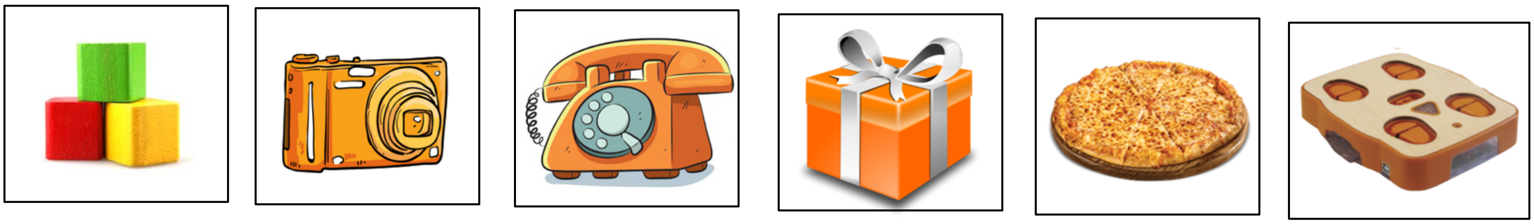
3. Circle the one that is most alike.

A KIBO and a computer	A KIBO and a pig
	
A KIBO and a bike	A KIBO and a house
	

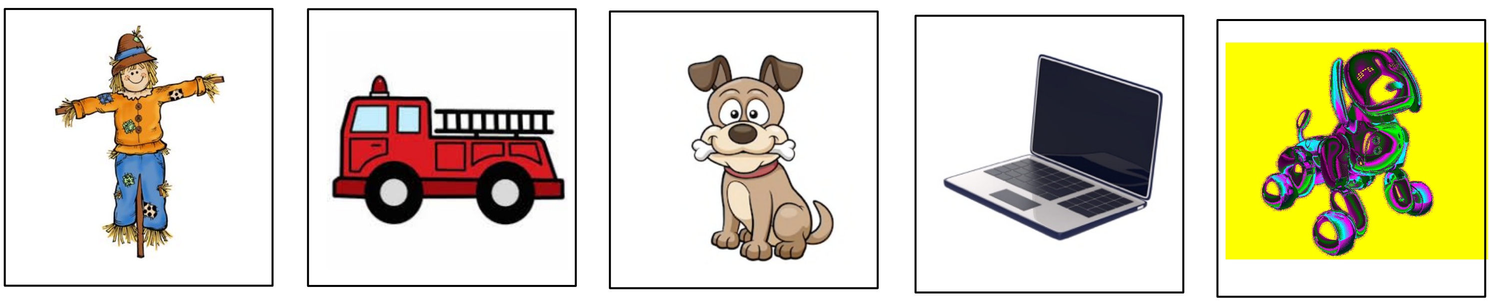
Name ______________________ Date ______________

Teacher ____________________

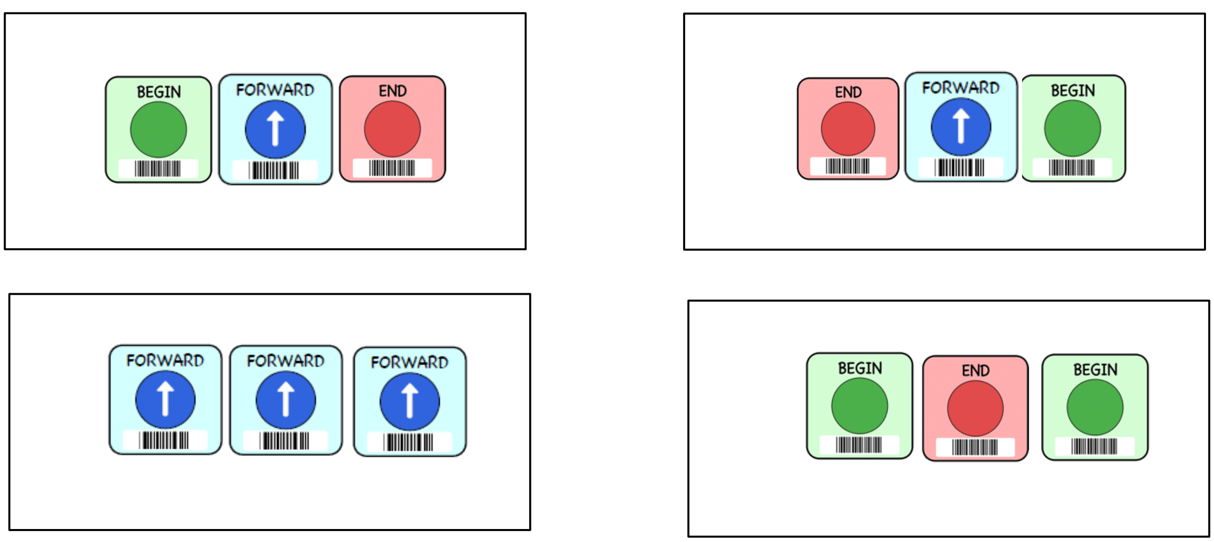
4. Circle one thing that is a part of a KIBO robot



5. Which one of these is a robot?



6. Which one of these will make a program that moves?



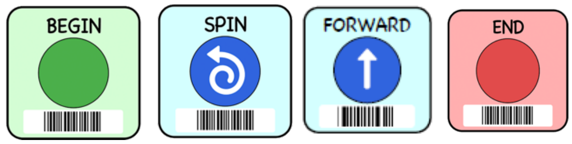
Appendix B-2. Solve-It Assessment B

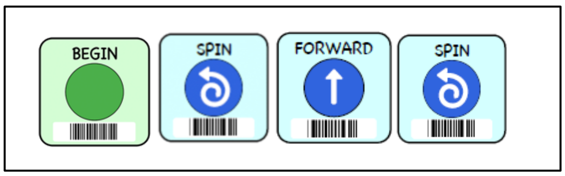
Name ______________________________ Date ___________________

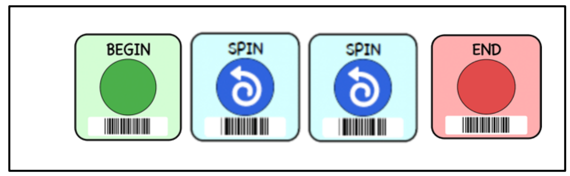
Teacher ________________________________

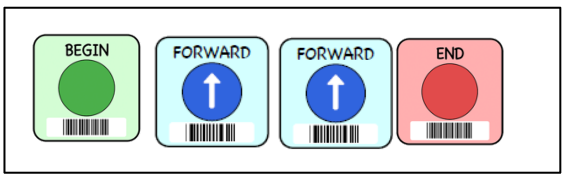
Circle the correct answers:

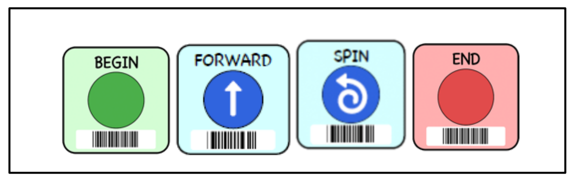
1. How can we change this program so that KIBO spins 2 times.



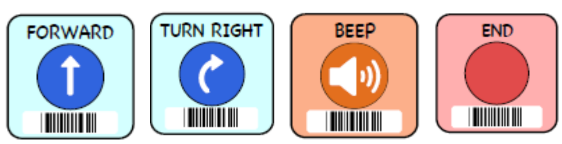




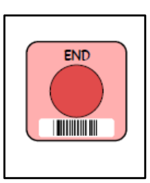




2. KIBO won't move forward, turn, or beep after I scan this program.



Circle one block that is needed to **Debug** this program:



3. Circle the one of these that allows KIBO to listen.





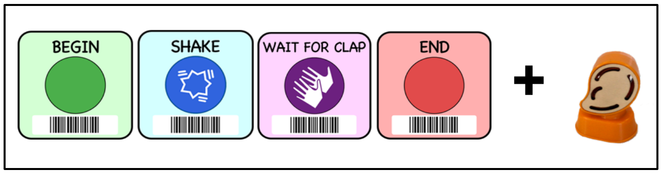
Solve-it

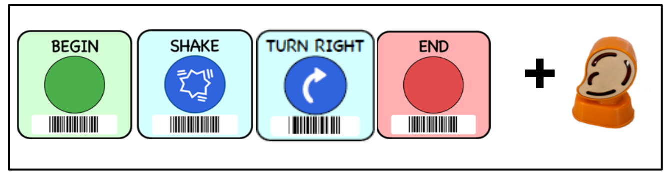
B

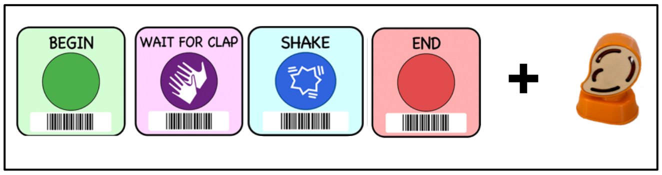
Name ______________________ Date ______________

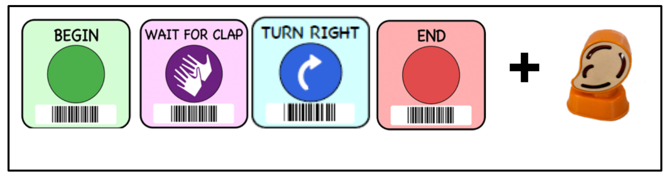
Teacher ________________________

4. Which of these programs will make KIBO shake after it hears a clap?

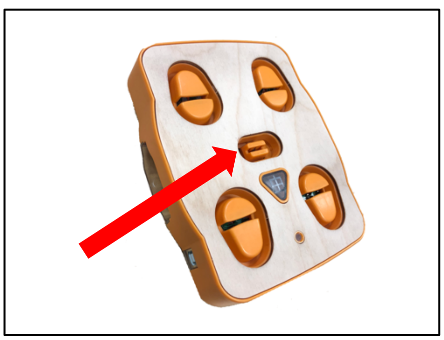


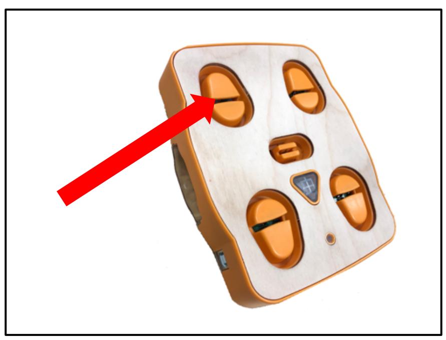


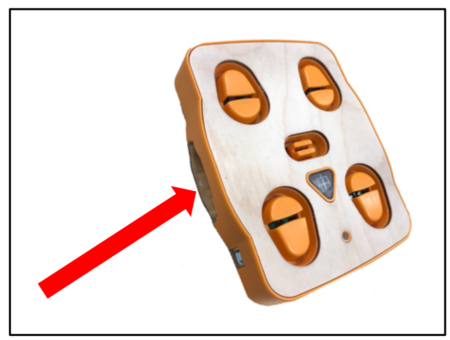




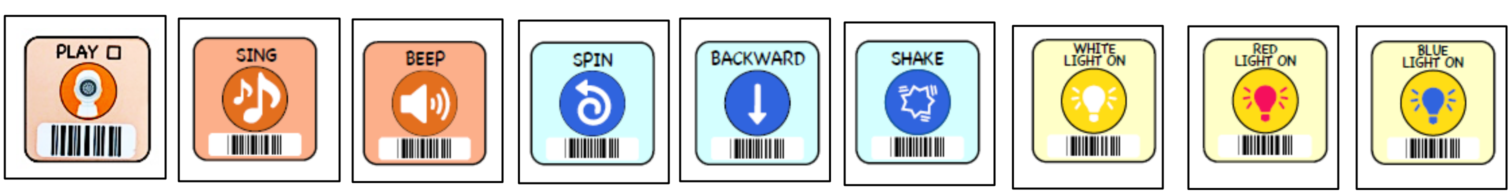
5. Where does This sound sensor go on KIBO?







6. Circle the block that would make this sensor play what you recorded?



Appendix B-3. Solve-It Assessment C

Solve-it

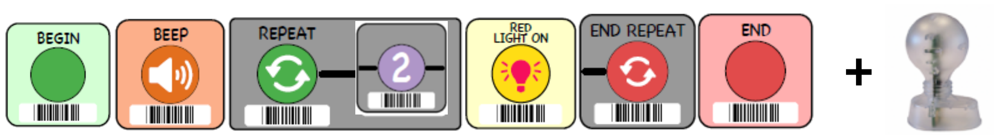
C

Name ______________________ Date _____________ 1

Teacher ____________________

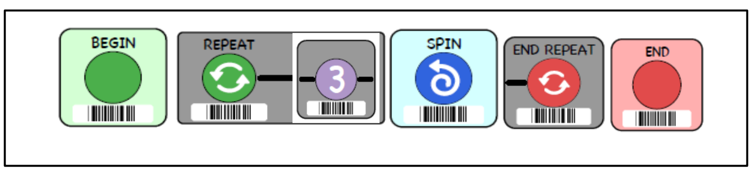
Circle the correct answer.

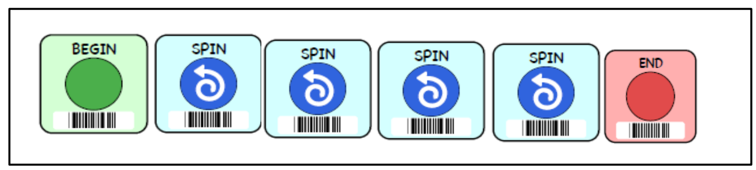
1. How many times will this program put a red light on?

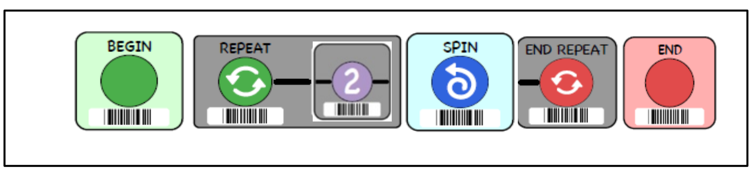


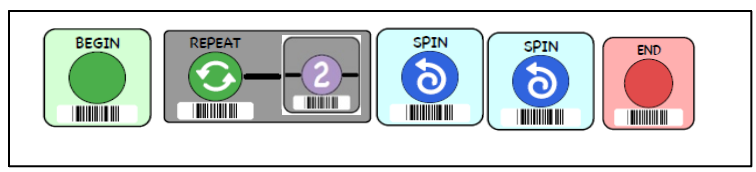
1	2	3	4

2. Which of these programs will make KIBO spin exactly 3 times?

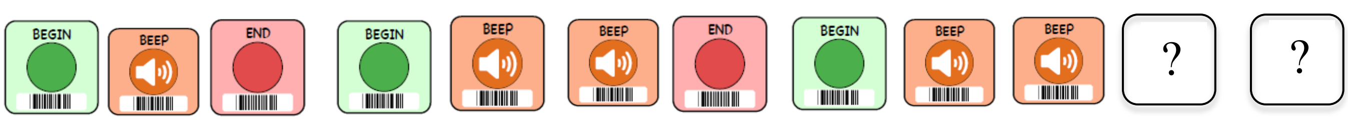


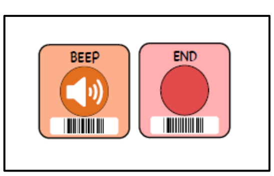


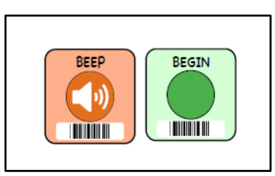


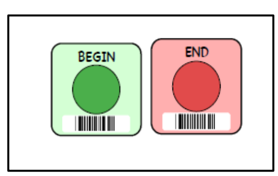


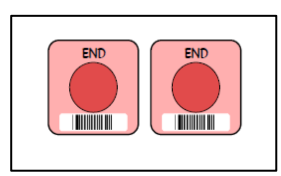
3. Which pair of blocks would complete this pattern?







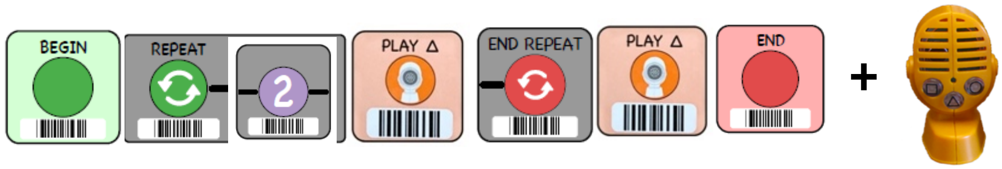




Name ______________________________ Date __________________

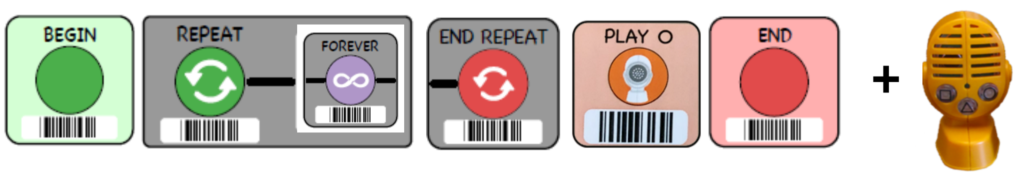
Teacher ___________________________

4. How many times will this program play the sound recording?

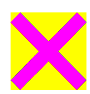


1	2	3	4

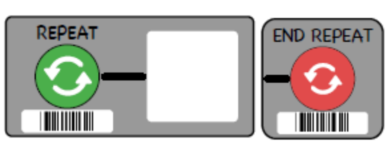
5. True or False: This program repeats the Play O sound forever.

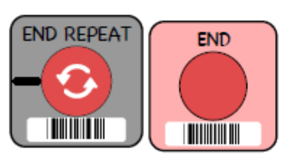


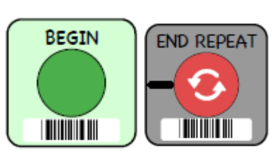
True 

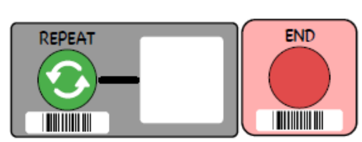
False 

6. All KIBO programs must have a BEGIN and END block. Which blocks must all repeat loops have?









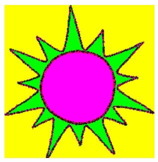
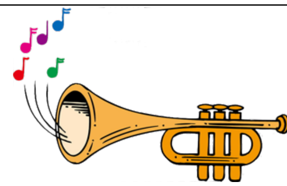
Appendix B-4. Solve-It Assessment D

Name ______________________ Date _____________

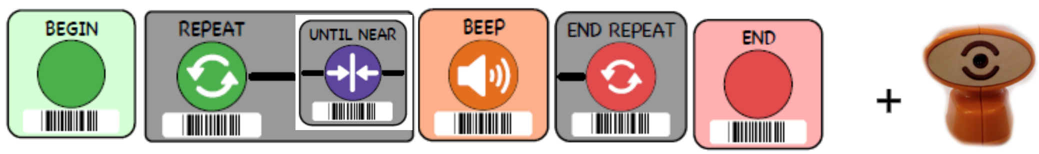
Teacher ____________________

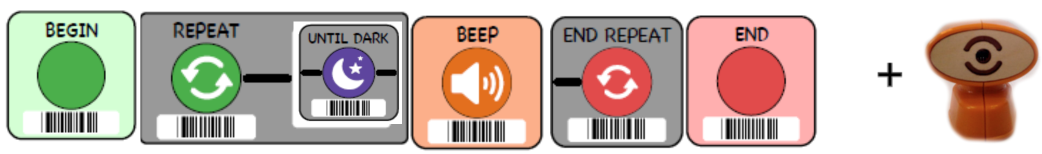
Circle the correct answer.

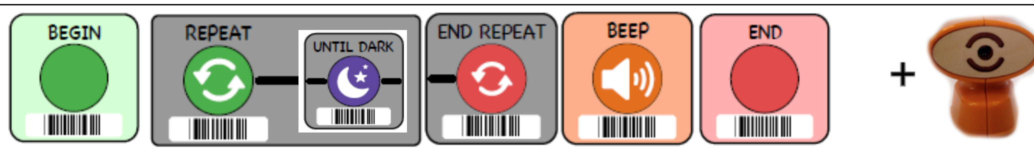
1. Which of these things does this sensor measure? 

 Light	 Sound	 Distance

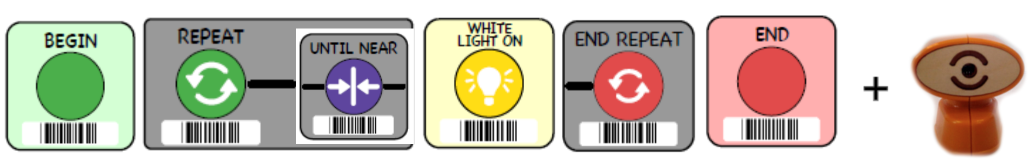
2. Which of these will keep beeping until the light is off?

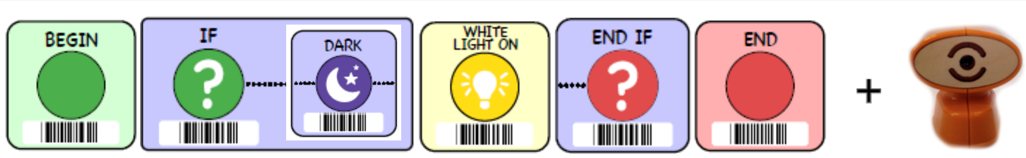






3. Which program will make KIBO only turn on its light if it is dark?

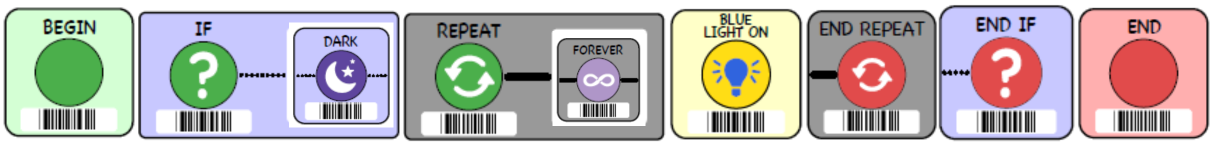


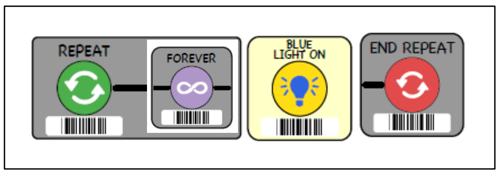


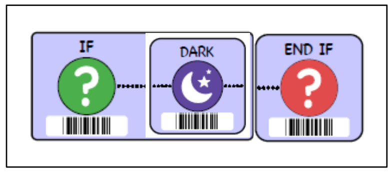
Name ______________________________ Date __________________

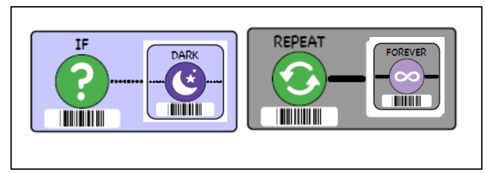
Teacher ___________________________

4. Which blocks are the **INNER** loop of this program?

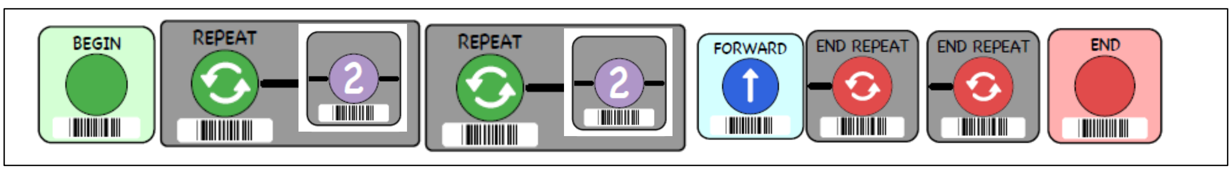


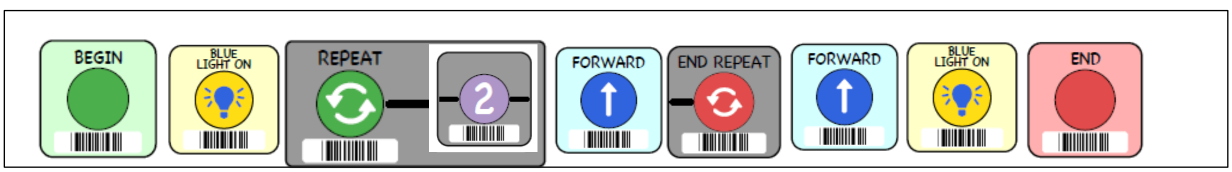


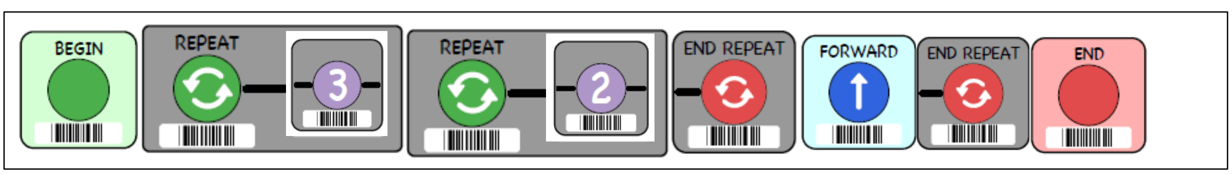




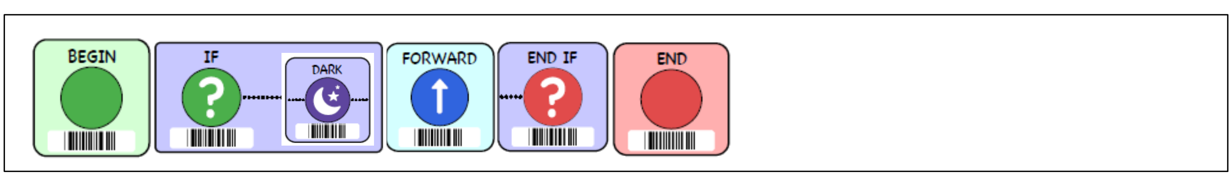
5. Which of these programs will move KIBO the most?

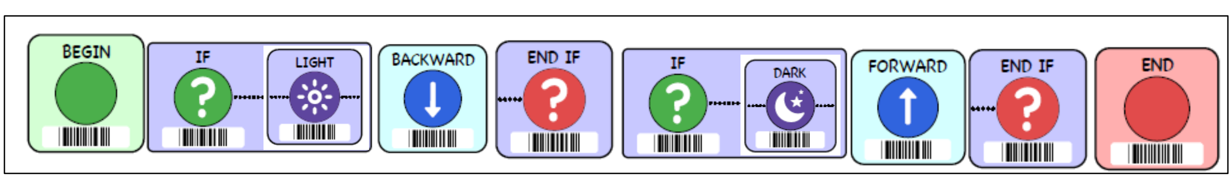


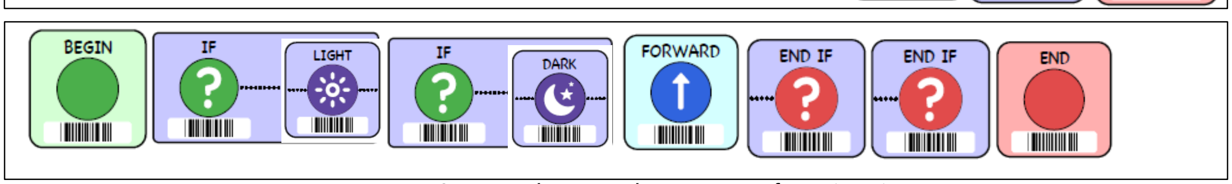




6. Which program will **NOT** move KIBO forward if it is dark?



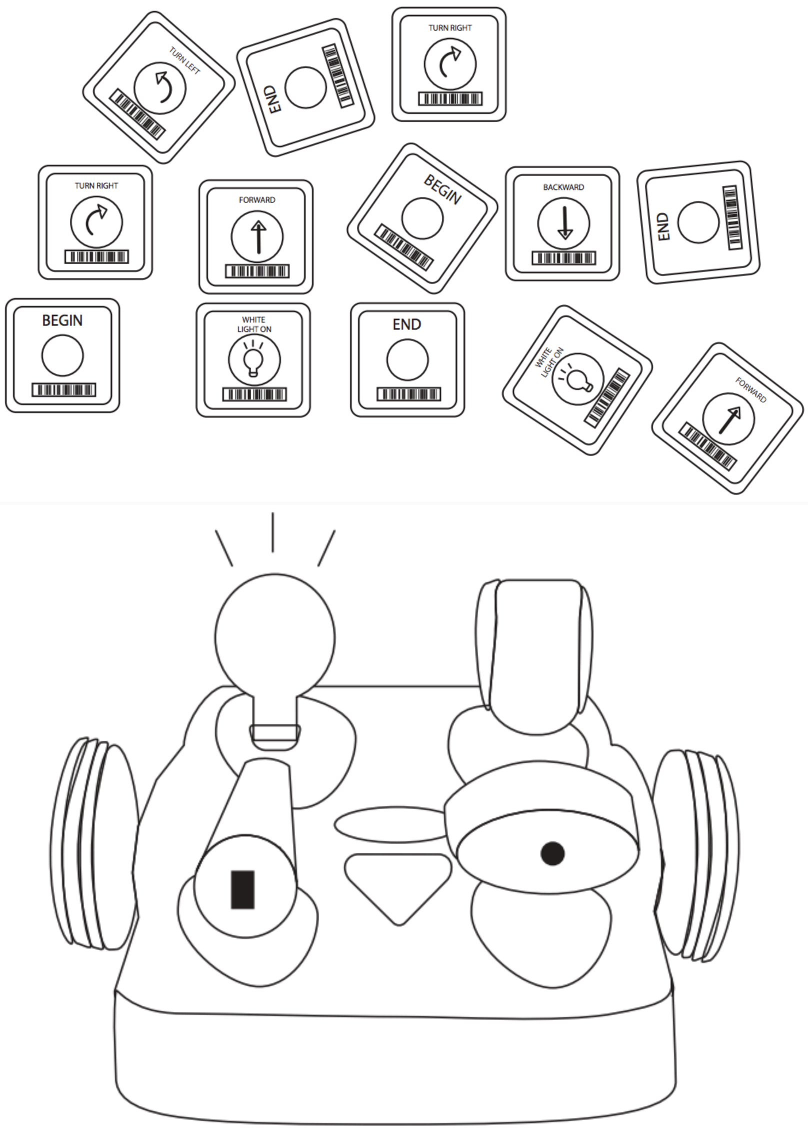




Appendix C. Design Journal

__________’s

Design Journal



Name: ____________________ Date: _____________________

Lesson 1:
How to Build a Robot

Describe your robot here.

____

____

____

____

____

____

____

What kinds of shapes are in your robot?

____

____

____

____

Name: ____________________ Date: _____________________

Lesson 1:
How to Build a Robot (continued)

Write instructions for how to build the robot that you designed. You can also draw pictures to help readers understand your directions.

____

____

____

____

____

____

____

____

Name: ____________________ Date: _____________________

Lesson 1:

Use this blank sheet for any extra planning or writing!

Name: ____________________ Date: _____________________

Lesson 2:

Use this blank sheet for any extra planning or writing!

Name: ____________________ Date: ____________________

Lesson 3:

Order Matters: Guide a Friend

Partner A: Write the steps below for Partner B to get to a certain "secret" spot in the room!

____

____

____

Have Partner B read your steps and follow the exact instructions.

CHECK! *Did Partner B reach your "secret" spot?*

Yes! Almost! Not at all!

Partner A: Rewrite the steps to make sure Partner B can make it to the correct spot!

____

____

____

CHECK AGAIN! *Did Partner B reach your "secret" spot?*

Yes! Almost! Not at all!

Name: ____________________ Date: _____________________

Lesson 3:

Use this blank sheet for any extra planning or writing!

Name: ____________________ Date: ____________________

Lesson 4:

Hokey-Pokey Reflection

Use the KIBO stickers to write your Hokey-Pokey program here.
Make sure the blocks are in the right order!

Which block did you use the most? ___________________________

How many times did you use this block? ______________________

Which block did you use the least? ___________________________

How many times did you use this block? ______________________

Name: ____________________ Date: _____________________

Lesson 4:

Use this blank sheet for any extra planning or writing!

Name: ____________________ Date: _____________________

Lesson 5:
Debugging Reflection

What was one problem you had with KIBO?

What were some things you tried to help solve the problem?

Which solution worked best?

Name: ____________________ Date: _____________________

Lesson 5:

Use this blank sheet for any extra planning or writing!

Name: ____________________ Date: _____________________

Lesson 6:

Use this blank sheet for any extra planning or writing!

Name: ____________________ Date: _____________________

Lesson 7:
If You're Wild and You Know It

Make a list of all the things you would want your robot to do to show that it is happy and wild!

1. ____

2. ____

3. ____

4. ____

5. ____

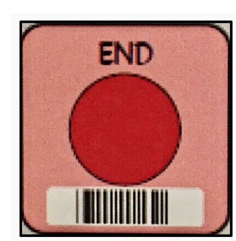
What blocks and modules would you use to program your robot to do these happy and wild things? Circle them:

Name: ____________________ Date: ____________________

Lesson 7:

If You're Wild and You Know It (continued)

Use the KIBO stickers to write the blocks of your final program here:



Name: ____________________ Date: _____________________

Lesson 7:

Use this blank sheet for any extra planning or writing!

Name: ____________________ Date: _____________________

Lesson 8:
Toothbrush Exercise

When you brush your teeth, what things do you do only one time?

____

____

When you brush your teeth, what things do you do more than one time?

____

____

Write instructions for how to brush your teeth. Make sure you have covered all the steps!

____

____

____

____

____

____

Name: ____________________ Date: _____________________

Lesson 8:

Use this blank sheet for any extra planning or writing!

Name: ____________________ Date: _____________________

Lesson 9:

My Five Senses

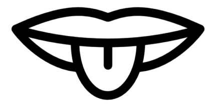
Choose one of the five senses below. Draw a circle around the sense that you have chosen.

See

Hear

Smell

Taste



Touch

Describe a time of when you last used this sense:

____

____

____

____

____

____

____

Name: ____________________ Date: _____________________

Lesson 9:

Use this blank sheet for any extra planning or writing!

Name: ____________________ Date: _____________________

Lesson 10:
Writing an Alternative Story

Write your own ending to *Where the Wild Things Are*! What would you want to happen differently in the story?

Use the space below to draft your alternative story:

____

____

____

____

____

____

____

____

____

____

____

____

Name: ____________________ Date: _____________________

Lesson 10:

Use this blank sheet for any extra planning or writing!

Name: ____________________ Date: _____________________

Lesson 11:
Wild Rumpus Composition

Describe your Wild Rumpus on the lines below. Make sure to cover these important details!

- Who is there?
- What does it look like?
- When does each activity happen?
- Where does it happen?
- Why did you choose these activities?

Name: ____________________ Date: _____________________

Lesson 11:
Coding the Wild Rumpus

Now that you have written about your Wild Rumpus, start planning your KIBO Wild Rumpus!

Ask: What activities do you want your robot to do in your Wild Rumpus?

____

____

____

____

Imagine: What will your robot look like?

My robot is a(an) _______________________________________

Its name is ____

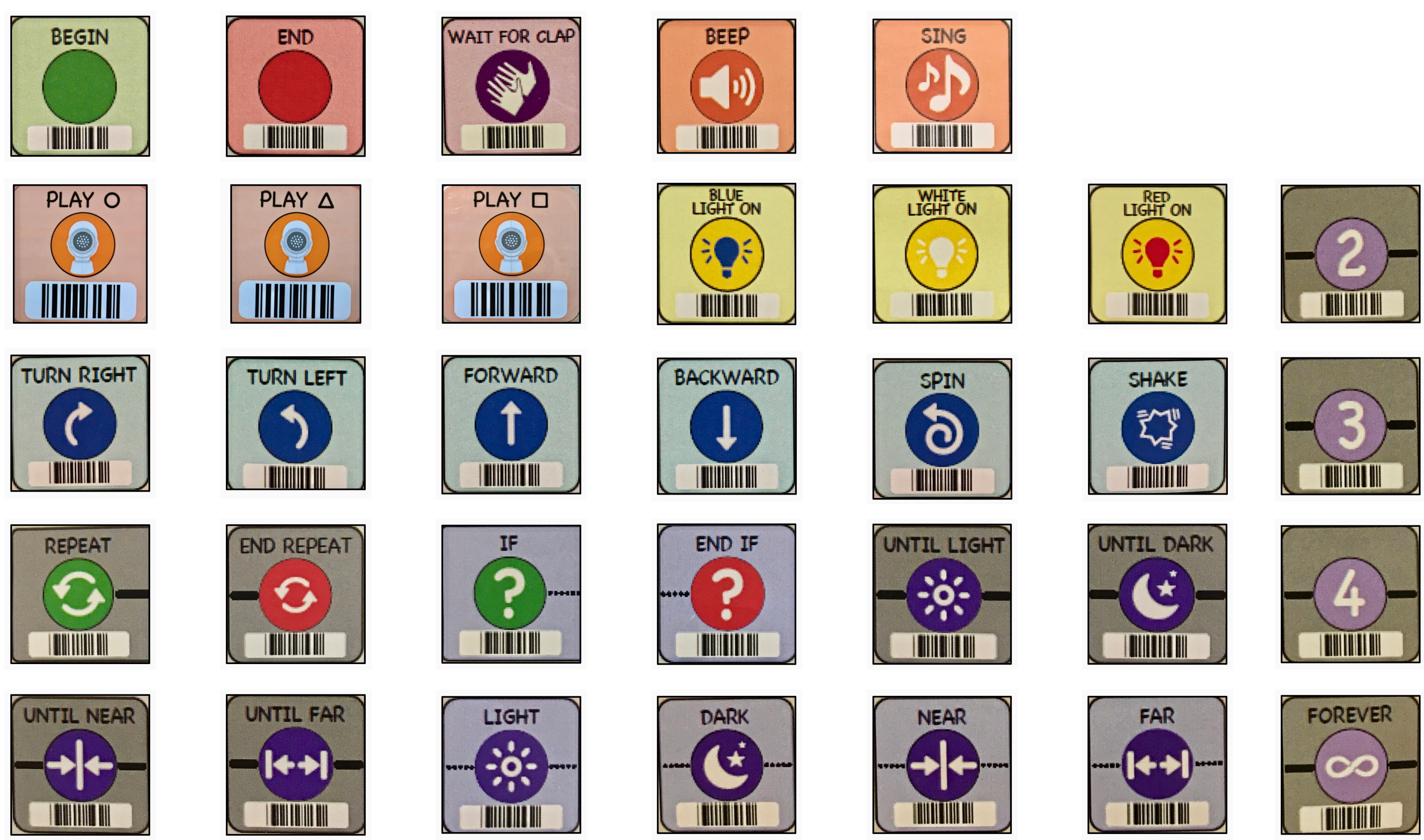
Draw what your robot will look like:

Name: ____________________ Date: ____________________

Plan: Which KIBO parts and blocks will you need to create your project?

Circle the KIBO parts you think you will need:

Circle the programming blocks you think your robot's program will need:



Before moving onto the next step, make sure to get feedback on your plan!

Name: ____________________ Date: _____________________

Create: **Gather your materials and programming blocks and get to work, engineer!**

Describe the kinds of materials you used to decorate your KIBO:

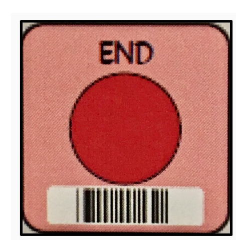
____

____

____

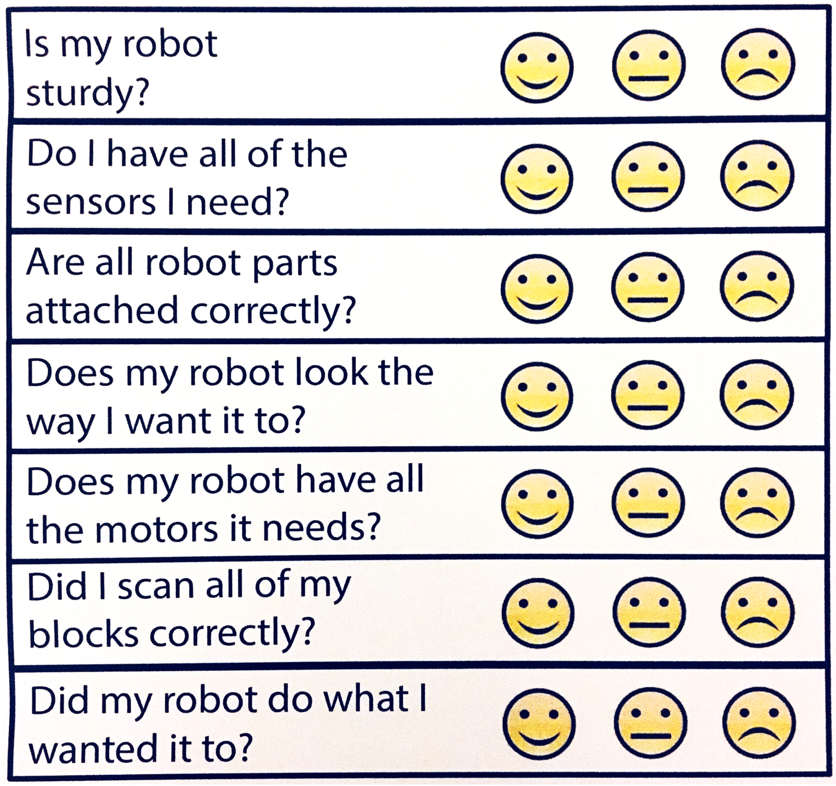
____

Use the KIBO stickers to write your robot's first program here. Don't worry if your first program doesn't work the way you want it to. You can always go back and make changes!



Name: ____________________ Date: _____________________

Test and Improve: Before engineers finish a project, they need to test and improve their work. Use this checklist to see how your robot is coming along!



Now it's time to improve and fix your "bugs"! Feel free to make as many changes as you want to improve your program or your robot's decorations.

What changes did you make to your project?

____

____

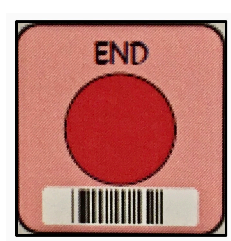
____

____

Name: ____________________ Date: ____________________

Share: Now that you have completed your robot, you can share what you have learned and achieved with friends, family, and other engineers.

Use the KIBO stickers to record your robot's final program:



Name: ____________________ Date: _____________________

Describe your final KIBO project. What did your robot look like? What kinds of decorations did you use?

What blocks and modules did you use in your program? How did using those parts make your Wild Rumpus come to life?

What was your **favorite** thing about working on your robot?

What was the **hardest** part about working on your robot?

Name: ____________________ Date: _____________________

Lesson 11:

Use this blank sheet for any extra planning or writing!

Name: ____________________ Date: _____________________

Lesson 12:

Use this blank sheet for any extra planning or writing!

Appendix D. Collaboration Web

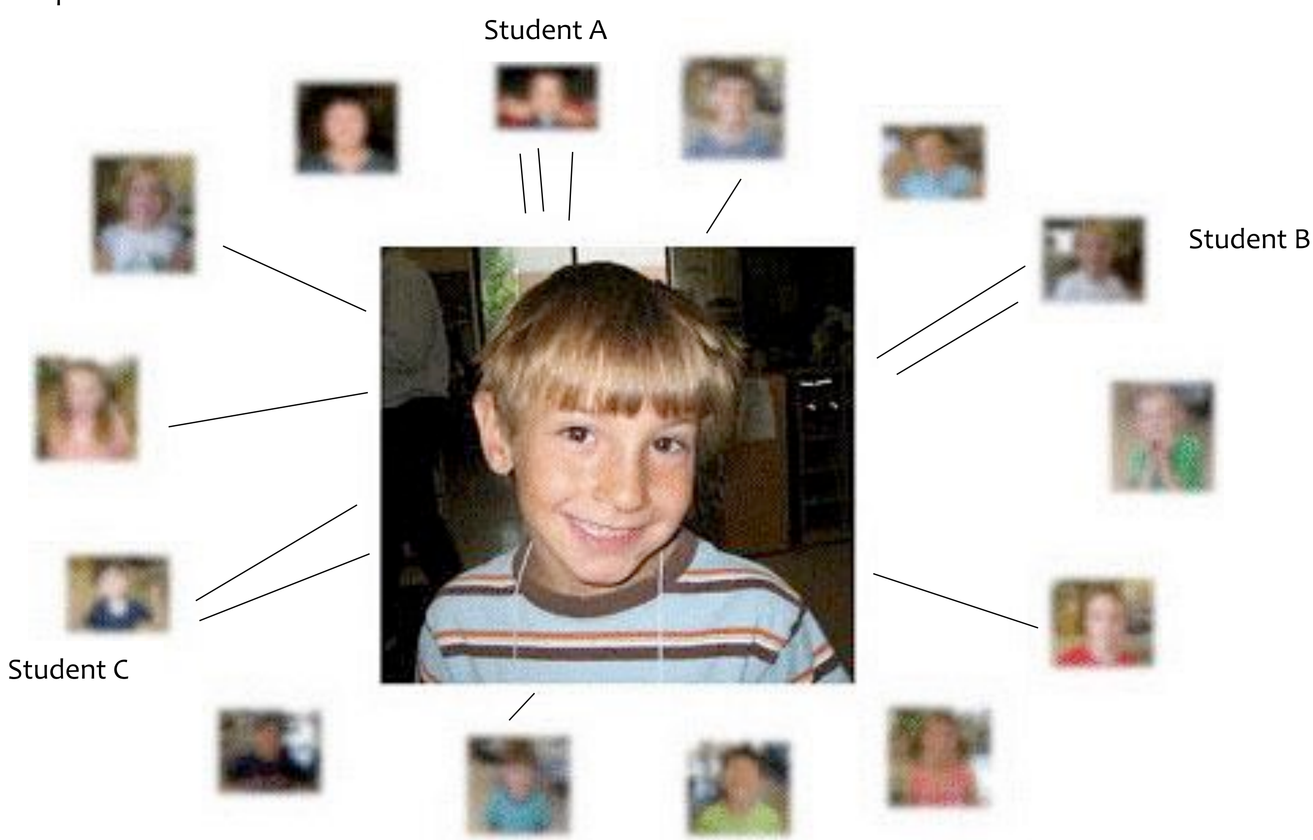
Appendix D. Collaboration Web

A collaboration web is a tool for students to recognize peers who have helped and supported them in different ways, such as working together on a common task, lending or borrowing materials, programming together, etc. Students will create a Collaboration Web during Lesson 8: The Wild Rumpus Project and write thank you letters to the three peers with whom they have collaborated the most.

Directions:

1. Obtain headshots of each student in the class.
2. Create individual printouts with each student's photograph in the center of the page and the names and photographs of all the other students arranged in a circle surrounding the central photo.
3. Whenever you observe students collaborating during the final project, ask students to draw a line from their photo in the center to the photo of the other students with whom they collaborated.
4. At the end of Lesson 8, ask students to count the number of lines they have with each student. Ask students to write thank you letters to the three students who have the most lines drawn to their photos.

Sample Collaboration Web:



The student in the center will write thank you letters to Students A, B, and C.

References

Bers, M. U. (2008). *Blocks to robots: Learning with technology in the early childhood classroom*. New York, NY: Teachers College.

Bers, M. U. (2012). *Designing digital experiences for positive youth development: From playpen to playground*. Cary, NC: Oxford.

Bers, M. U. (2018). *Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom*. New York, NY: Routledge Press.

Darragh, J. C. (2006). *The environment as the third teacher*.
Retrieved from http://www.eric.ed.gov/PDFS/ED493517.pdf

International Society for Technology in Education (2017). *ISTE Standards for Students*.
Retrieved from https://www.iste.org/standards/for-students

K–12 Computer Science Framework. (2016). Retrieved from http://www.k12cs.org.

Massachusetts Department of Education (2016). *Digital Literacy and Computer Science Framework*. Retrieved from https://www.doe.mass.edu/frameworks/dlcs.pdf

National Governors Association Center for Best Practices & Council of Chief State School Officers

(2010). *About the Standards*. Retrieved from http://www.corestandards.org/about-the-standards/

Papert, S. (1993). *Mindstorms: Children, computers, and powerful ideas*. New York, NY: Basic Books.

Virginia Department of Education (2017). *Computer Science Standards of Learning* (SOL). Retrieved from http://www.doe.virginia.gov/testing/sol/standards_docs/computer-science/index.shtml

Vygotsky, L. S. (2012). *Thought and Language*. Cambridge, MA: The MIT Press.