A KIBO Coding Curriculum for Emergent Readers

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

Tufts University | DEVTECH RESEARCH GROUP

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 *There Was an Old Lady Who Swallowed a Fly* by Simms Taback, which about an old lady who accidentally swallows a fly and then swallows other animals in order to try to retrieve it.

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
- 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 build their skills to 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? (10 min) • The Design Process (10 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	• There Was an Old Lady Who Swallowed a Fly (20 min) • Order Matters (15 min) • Program the Teacher with KIBO Blocks (10 min) • Meet the KIBO Robot (15 min)
Lesson 4: Taking Care of Our Materials	• Lesson Introduction (5 min) • How to Treat Our Materials (15 min) • Conventions of KIBO Usage (15 min) • Procedure Practice (10 min) • Left to Right on Paper and KIBO (15 min)
Lesson 5: Programmer and Author	• Lesson Introduction (5 min) • There Was an Old Lady Who Swallowed a Fly (15 min) • Beginning, Middle and End (20 min) • Be a Programmer (20 min)
Lesson 6: Programming	• Coding with KIBO (10 min) • Dance the Hokey-Pokey (5 min) • Program the Hokey-Pokey (20 min) • Hokey-Pokey Reflection (5 min) • Share Creations (10 min) • Solve-It Assessment A (10 min)
Lesson 7: Debugging	• Why is KIBO Confused? (20 min) • Free Play (20 min) • Debugging Reflection (20 min)
Lesson 8: Cause and Effect	• Exploring Our Senses (10 min) • KIBO Sound Sensor (5 min) • Free Play (15 min) • Reflection (10 min) • Share Creations (10 min) • Solve-It Assessment B (10 min)
Lesson 9: Repeat Loops	• Repetition in Stories and Songs (15 min) • Toothbrush Exercise (15 min) • KIBO Repeat with Numbers (15 min) • Free Play (15 min)
Lesson 10: Final Project - Characterization	• There Was an Old Lady Who Swallowed a Fly (10 min) • Design A Character (10 min) • Create Your Own Character (20 min) • Character Share (10 min) • Solve-It Assessment C (10 min)
Lesson 11: Final Project - Retelling	• Reread (10 min) • Sequencing Animals (5 min) • Begin Coding the Retell (30 min) • Share Projects (15 min)

Lesson 12: Final Project - Expansion	• Treading Along a Familiar Path (10 min) • Design an Expansion (10 min) • Program the Story (20 min) • Share Creations (10 min) • Solve-It Assessment (10 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

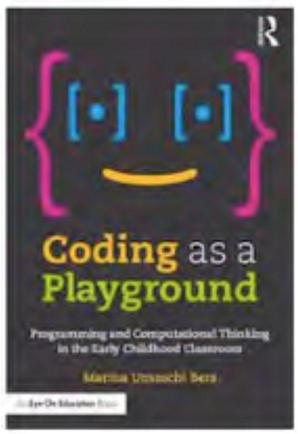
21 programming blocks

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 in groups on small class projects, exploring with each other during free play, or on their final KIBO projects, children will constantly be given a chance to collaborate while working with KIBO.

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. The recommended ratio is 1 KIBO per 2-4 children, so that all children have a chance to interact with KIBO. 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. To facilitate teamwork and equal participation, Appendix A includes examples of Job Cards that can be distributed to the children, which will assign specific roles to each child when working with KIBO.

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 kindergarten 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 *There Was an Old Lady Who Swallowed a Fly*) 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). Lessons in this curriculum are also aligned with the nationally recognized computer science frameworks **K–12 Computer Science Framework** (2016).

Table 2: Alignment of Standards

	Powerful Ideas of Computational Thinking (CT) and Literacy Embedded in Each Lesson	**Common Core ELA/ Literacy Framework (Kindergarten)**	**Computer Science Framework Alignment** (Based on the "by end of Grade 2 band")
1: Foundations	***CT***: *Design Process* ***Literacy***: *Writing Process*	**CCSS.ELA-LITERACY.W.K.5** With guidance and support from adults, respond to questions and suggestions from peers and add details to strengthen writing as needed.	**Algorithms and Programming:** Algorithms: People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow.

2: Technological Tools - Robots	***CT**: Hardware/Software, Representation* ***Literacy**: Tools of Communication*	**CCSS.ELA-LITERACY.W.K.6** With guidance and support from adults, explore a variety of digital tools to produce and publish writing, including in collaboration with peers.	**Computing Systems** Hardware and Software: A computing system is composed of hardware and software. Hardware consists of physical components, while software provides instructions for the system. These instructions are represented in a form that a computer can understand. Devices: People use computing devices to perform a variety of tasks accurately and quickly. Computing devices interpret and follow the instructions they are given literally.
3: Sequencing	***CT**: Hardware/Software, Algorithms, Representation* ***Literacy**: Summarizing/Retelling the Sequence of a Story, Descriptive Language in Writing*	**CCSS.ELA-LITERACY.W.K.3** Use a combination of drawing, dictating, and writing to narrate a single event or several loosely linked events, tell about the events in the order in which they occurred, and provide a reaction to what happened. **CCSS.ELA-LITERACY.RL.K.2** With prompting and support, retell familiar stories, including key details.	**Algorithms and Programming:** Algorithms: People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow. Control: Computers follow precise sequences of instructions that automate tasks. Program execution can also be nonsequential by repeating patterns of instructions and using events to initiate instructions.
4: Taking Care of Our Materials	***CT**: Hardware/Software* ***Literacy**: Book Handling*		**Computing Systems** Hardware and Software: A computing system is composed of hardware and software. Hardware consists of physical components, while software provides instructions for the system. These instructions are represented in a form that a computer can understand.

5: Programmer and Author	***CT**: Design Process, Algorithms, Control Structures* ***Literacy**: Writing Process, Sequence*	**CCSS.ELA-LITERACY.W.K.5** With guidance and support from adults, respond to questions and suggestions from peers and add details to strengthen writing as needed. **CCSS.ELA-LITERACY.W.K.3** Use a combination of drawing, dictating, and writing to narrate a single event or several loosely linked events, tell about the events in the order in which they occurred, and provide a reaction to what happened.	**Algorithms and Programming:** Program Development: People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems.
6: Programming	***CT**: Algorithms, Design Process* ***Literacy**: Descriptive Language in Writing*	**CCSS.ELA-LITERACY.W.K.2** Use a combination of drawing, dictating, and writing to compose informative/explanatory texts in which they name what they are writing about and supply some information about the topic.	**Algorithms and Programming:** Program Development: People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems.
7: Debugging	***CT**: Debugging* ***Literacy**: Editing, Awareness of Audience*	**CCSS.ELA-LITERACY.W.K.5** With guidance and support from adults, respond to questions and suggestions from peers and add details to strengthen writing as needed.	**Computing Systems** Troubleshooting: Computing systems might not work as expected because of hardware or software problems. Clearly describing a problem is the first step toward finding a solution.

8: Cause and Effect	***CT**: Control Structures, Representation, Sensors* ***Literacy**: Spelling-Sound Correspondence*	**CCSS.ELA-LITERACY.RF.K.3.A** Demonstrate basic knowledge of one-to-one letter-sound correspondences by producing the primary sound or many of the most frequent sounds for each consonant.	**Algorithms and Programming:** Algorithms: People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow. Control: Computers follow precise sequences of instructions that automate tasks. Program execution can also be nonsequential by repeating patterns of instructions and using events to initiate instructions. Variables: Information in the real world can be represented in computer programs. Programs store and manipulate data, such as numbers, words, colors, and images. The type of data determines the actions and attributes associated with it.
9: Repeat Loops	***CT**: Control Structure, Modularity* ***Literacy**: Repetition as a Literary Device, Repetition in Word Forms*	**CCSS.ELA-LITERACY.RL.K.5** Recognize common types of texts (e.g., storybooks, poems).	**Algorithms and Programming** Modularity: Complex tasks can be broken down into simpler instructions, some of which can be broken down even further. Likewise, instructions can be combined to accomplish complex tasks.
10: Final Project - Characterization	***CT**: Representation* ***Literacy**: Character, Point of View*	**CCSS.ELA-LITERACY.RL.K.9** With prompting and support, compare and contrast the adventures and experiences of characters in familiar stories.	**Algorithms and Programming** Variables: Information in the real world can be represented in computer programs. Programs store and manipulate data, such as numbers, words, colors, and images. The type of data determines the actions and attributes associated with it.

11: Final Project - Retelling	***CT**: Representation, Algorithms, Sequence* ***Literacy**: Character, Point of View, Summarizing/Retelling the Sequence of a Story*	**CCSS.ELA-LITERACY.W.K.3** Use a combination of drawing, dictating, and writing to narrate a single event or several loosely linked events, tell about the events in the order in which they occurred, and provide a reaction to what happened. **CCSS.ELA-LITERACY.RL.K.2** With prompting and support, retell familiar stories, including key details.	**Algorithms and Programming:** Algorithms: People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow. Control: Computers follow precise sequences of instructions that automate tasks. Program execution can also be nonsequential by repeating patterns of instructions and using events to initiate instructions. Variables: Information in the real world can be represented in computer programs. Programs store and manipulate data, such as numbers, words, colors, and images. The type of data determines the actions and attributes associated with it.
12: Final Project - Expansion	***CT**: Design Process* ***Literacy**: Writing Process*	**CCSS.ELA-LITERACY.W.K.5** With guidance and support from adults, respond to questions and suggestions from peers and add details to strengthen writing as needed.	**Algorithms and Programming:** Algorithms: People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow.

Lesson 1: Foundations

Powerful Idea From Computer Science:	**Powerful Idea From Literacy:**
Design Process	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? (10 min)
- The Design Process (10 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 the Design Process that engineers go through
- 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*

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? (10 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.

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.

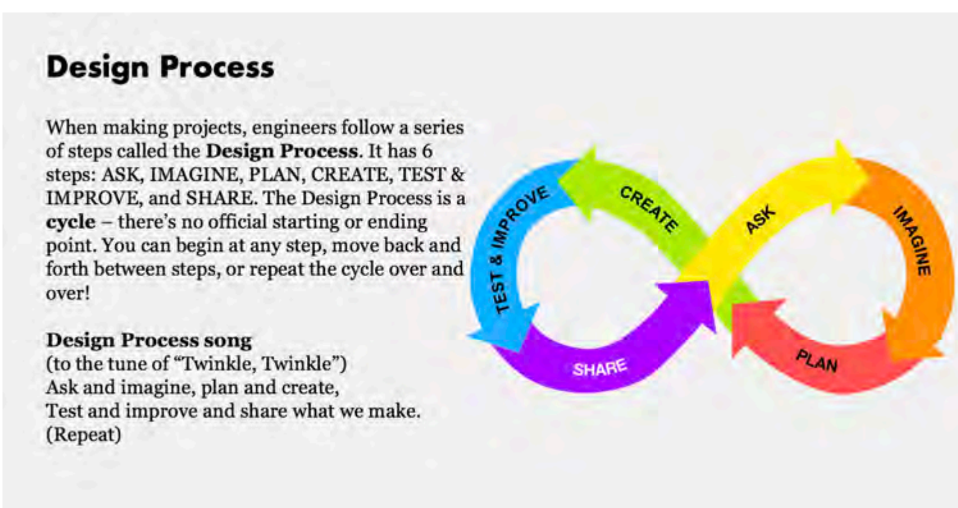
For descriptions and further activity ideas, check out the following resources:

- 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

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?*

THE DESIGN PROCESS (10 min)

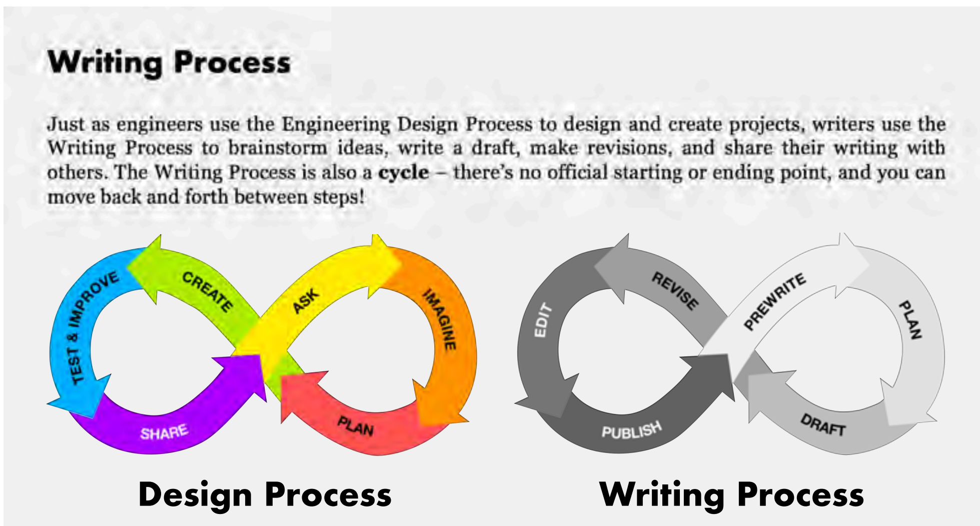
Explain to students that in order for engineers to build solutions for problems, they need to go through the steps of the Design Process. To help them understand the Design Process better, give the students a problem for them to solve.
Ask students: If there's a bug on the ceiling, how can I reach the bug and catch it?
Help the preschoolers imagine solutions by suggesting materials you can use in the classroom - a chair, cup, paper - to introduce them to ways the problem can be addressed. As they walk through imagining solutions and creating a plan, let them know that they just began the steps of Design Process. Continue to walk through the rest of the steps of the Design Process using the example you have chosen. Engage with the preschoolers through a design process song detailing each stage of the series.



ENGINEERS AND WRITERS (10 min)

Ask students: What are you favorite books?

Explain to students similarities between the design process and the writing process. They are both 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.



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. Here, each step in a how-to book should be a sketch or picture, with an accompaniment of words depending on the students' writing level.

Pass out the Design Journals. Ask students to create a "How-To Book" for building their own robot. Read each aspect of the Design Journal aloud to the students. Ask students to draw as many pictures of their steps as possible so that someone else can learn how to build their robot simply by looking at their instructions.

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 pictures 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*
- An image of a cow*

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 picture of a cow. Have a few students say out loud what is in the picture. *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 pictographs 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.*
> *Peter never stops talking.*

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

- There Was an Old Lady Who Swallowed a Fly (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
- ☐ 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 *There Was an Old Lady Who Swallowed a Fly* by Simms Taback
- 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

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

THERE WAS AN OLD LADY WHO SWALLOWED A FLY (20 min)

Read the book *There Was an Old Lady Who Swallowed a Fly* 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 she swallowed a fly, then a spider to catch the fly, etc.). *Then ask students: What if the first scene was the old lady swallowing the spider? How would that change the story? What would she swallow next? Would she still swallow the fly?* The purpose of this activity is to get students to think about sequencing in narrative.

ORDER MATTERS (15 min)

Tell the students that just like order mattered in the book, order matters in programming as well. Programs are instructions that are followed by computers or machines. We all use instructions (use whatever word your students are familiar with i.e. directions, procedures, etc.) every day, both here at school and when you're at home.

Ask students: What are some instructions that we have to follow here at school?

> *Example: How do we line up at the door for lunch? First, you have to stand up. Then, you have to push in your chair. Then, you have to walk to the line. Would you push in your chair before standing up?*

Ask students: Who can think of other step by step instructions that we use at school? What about at home?

> *Examples: Brushing your teeth, tying shoes.*

Conclude the activity by bringing it back to how computers have to follow instructions as well - reflect on the importance of sequencing in literacy and computer science. *Ask students: What would happen if you tried to line up for lunch but didn't stand up first? What would happen if you didn't put toothpaste on your toothbrush until after you brushed your teeth? Why did the order matter in each activity?*

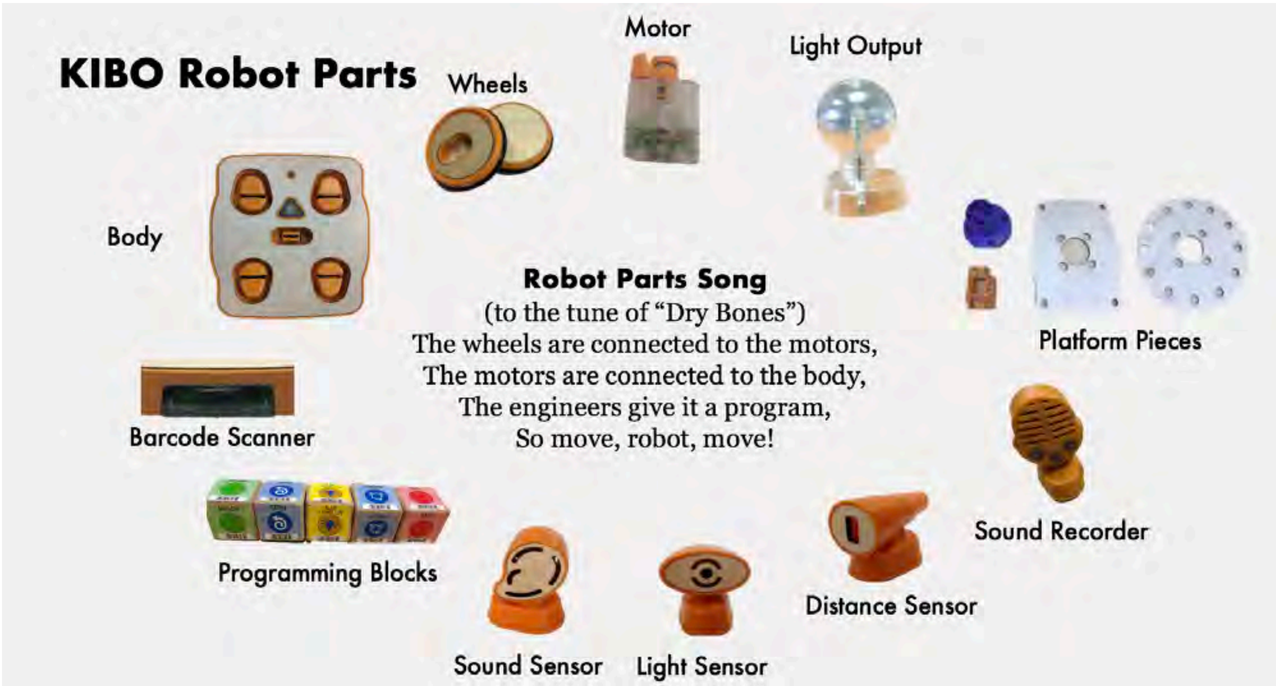
PROGRAM THE TEACHER WITH KIBO BLOCKS (10 min)

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.


KIBO Robot Parts
Wheels
Motor
Light Output
Body
Robot Parts Song
(to the tune of "Dry Bones")
The wheels are connected to the motors,
The motors are connected to the body,
The engineers give it a program,
So move, robot, move!
Platform Pieces
Barcode Scanner
Sound Recorder
Programming Blocks
Distance Sensor
Sound Sensor
Light Sensor

Lesson 4: Taking Care of Our Materials

Powerful Idea From Computer Science:
Hardware/Software

Powerful Idea From Literacy:
Book Handling

OVERVIEW

This lesson will introduce concepts of taking care of our materials, with a focus on books and KIBO parts. Activities in this lesson will also cover important text and program conventions such as going from left to right, holding materials right side up and being gentle with handling supplies.

PURPOSE

The activities set up an important foundation for how students handle their materials as they engage with books and KIBO. They also introduce conventions of literacy and programming.

ACTIVITIES

- Lesson Introduction (5 min)
- How to Treat Our Materials (15 min)
- Conventions of KIBO Usage (15 min)
- Procedure Practice (10 min)
- Left to Right on Paper and KIBO (15 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
- ☐ 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
- ☐ Create anchor chart for the How to Treat Our Materials activity*

MATERIALS

FOR THE TEACHER:

- Anchor chart for the How to Treat Our Materials Activity*
- 1 flathead screwdriver
- Extra AA batteries

FOR STUDENTS:

- 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

LESSON INTRODUCTION (5 min)

Explain to the students how this lesson is going to talk about treating our books and KIBO with respect and kindness, just like we treat each other with respect and kindness.

Ask students: Who has seen a book with ripped pages, missing pages or stains or marks?

Feel free to show examples if you have books in your class that have been treated poorly. Discuss why it's important to maintain our materials - *if we rip books, you may not get to use them again, or our class might run out. It's the same way with KIBO. We want to treat our KIBOs the same way we treat our books OR be even more careful with them.*

HOW TO TREAT OUR MATERIALS (15 min)

As a class, create an anchor chart utilizing pictures that shows good and bad ways to treat books and KIBO. Come up with some classroom procedures surrounding KIBO usage. Examples:

- Use gentle hands with KIBO and books
- Always carry with two hands if you are walking around
- Do not run with KIBO parts
- Put KIBO parts away at snack time

CONVENTIONS OF KIBO USAGE (15 min)

This activity will be a side by side comparison of a book and KIBO to cover topics such as left to right and right side up. Feel free to incorporate strategies or concepts your students have been working on in literacy. Have student give up a thumbs up when something is correct and a thumbs down if something is incorrect. Examples:

- I should hold my KIBO like this (upside down).
- I should hold my KIBO like this (vertical).
- When I'm done with my KIBO I should toss it onto the ground!
- If I get mad, I should slam my KIBO onto the table!
- I should walk very carefully with my KIBO held in two hands.
- I should drink my milk and eat my snack right next to my KIBO!

PROCEDURE PRACTICE (10 min)

This is an opportunity for you to practice with your students any procedures in your classroom related to KIBO usage. You may want to have students practice retrieving their KIBO from storage and bringing to their desk, practice putting away KIBO and its parts, how to properly let the instructor know when KIBO is running low on batteries, etc. You may want to practice how you will get students' attention when you need to make an announcement while they are playing with KIBO. We recommend a call to action that requires students to place both hands in the air.

LEFT TO RIGHT ON PAPER AND KIBO (15 min)

There are many different ways to teach young readers that text is read from left to right. Teachers will most likely have this established in their classroom. Feel free to use your chosen method here. We want to make a connection that just as a book is read from left to right, programs will be too. We recommend doing quick read aloud and demonstrating intentionally with your finger that you start at the left and move toward the right. Then, show class an example of a KIBO program and state that it functions the same way. Teach students how to scan the barcodes using KIBO.

Lesson 5: Programmer and Author

Powerful Idea From Computer Science:
Design Process, Algorithms, Control Structures

Powerful Idea From Literacy:
Writing Process, Sequence

OVERVIEW

Previously, students learned about the different characteristics of robots and KIBO's programming language. This lesson will build upon the topic and help students further understand programming through comparing a programmer to an author.

PURPOSE

This lesson allows students to connect their understanding of an author with their understanding of a programming and draw parallels.

ACTIVITIES

- Lesson Introduction (5 min)
- There Was an Old Lady Who Swallowed a Fly (15 min)
- Beginning, Middle and End (20 min)
- Be a Programmer (20 min)

STUDENTS WILL BE ABLE TO...

- Understand parallels between authors and programmers
- Understand that both programs and stories need a logical order (beginning, middle, and end) to make sense

PREPARATION FOR TEACHERS

- ☐ Read through the Activity Guide
- ☐ Print picture of Simms Taback (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

MATERIALS

FOR THE TEACHER:

- 1 copy of *There Was an Old Lady Who Swallowed a Fly* by Simms Taback
- Picture of Simms Taback (optional)*
- 1 flathead screwdriver
- Extra AA batteries
- Appendix E for troubleshooting tips

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 example

VOCABULARY

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

Lesson 5 Activities

LESSON INTRODUCTION (5 min)

Ask students: Two classes ago, we learned about programs. Can anyone tell me what a program is? Now that we know what a program is, what do you think is a programmer? Do you know anyone who is a programmer? What kind of things do you think they do?

Let the students know that today, they will get to become programmers. But before they get to become programmers, talk to them about other people that get to create stories.

Ask students: Does anyone know what an author is? (This may be something your class has already covered - if so, this is a great way to check for understanding and make the comparison to a programmer.)

An author writes the words of a story just like a programmer writes step-by-step instructions.

THERE WAS AN OLD LADY WHO SWALLOWED A FLY (15 min)

Read the book *There Was an Old Lady Who Swallowed a Fly* as a class. Be sure to emphasize the author's name on the title page and cover. Reinforce by asking the students "Who is the author of *There Was an Old Lady Who Swallowed a Fly?*" with the sentence starter "The author is _______."

The author of *There Was an Old Lady Who Swallowed a Fly* is Simms Taback. Being the author of a book means creating and writing the words of the book. Let the students know that on top of becoming a programmer, they will get the chance to also be an author.

Optional: Project or print a picture of Simms Taback so that they see it is a real person. Oftentimes, kids have a difficult time making the jump to understanding that books are written by people!

BEGINNING, MIDDLE AND END (20 min)

Remind students that every program has a beginning, middle and end. Using the KIBO Says cards, make a simple program and have the students identify the beginning, middle and end. An example would be using these blocks: Begin, Forward, Spin, Forward, End. The beginning is the Begin block. The middle is the Forward, Spin, Forward blocks. The end is the End block.

Explain to the students that just like a program, a story also has a beginning, middle and end. Walk through the beginning, middle and end of *There Was an Old Lady Who Swallowed a Fly*.

Ask students: What happens at the beginning of There Was an Old Lady Who Swallowed a Fly? What happens in the middle? What happens in the end? What happens if we take out the beginning (or middle, or end)?

BE A PROGRAMMER (20 min)

Tell students that we are going to create our own story with KIBO. *We will make a program that tells a story using the KIBO programming blocks*. Remind students that they learned both programs and stories have to have a beginning, middle and end.

Demonstrate for the class: Just like every story has to start with a beginning, we have to start our programs with the begin block (show begin block). Then, just like a story, our program has to have a middle (add any blocks to the begin!). Then, to show that we're done with our program, we have to have an end block (dd end block).

Scan the blocks and play the program. Scan the blocks without the middle (or beginning, or end) to demonstrate to the students how without a beginning, middle and end, a program (or story) does not work.

Lesson 6: 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

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

STUDENTS WILL BE ABLE TO...

- Develop programs with a specific goal in mind
- Practice scanning programs with KIBO
- Learn strategies for debugging and editing

PREPARATION FOR TEACHERS

- ☐ Read through the Activity Guide
- ☐ Print Job 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
- ☐ Print Solve-It Assessment A (one for each student)

MATERIALS

FOR THE TEACHER:

- Job Cards*
- 1 flathead screwdriver
- Extra AA batteries
- Appendix E for troubleshooting tips

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 6: Activities

CODING WITH KIBO (10 min)

Ask students: Who remembers what programs mean?

Remind the students about programs. Programs are instructions that are followed by computers or machines. We all use *instructions* (use whatever word your students are familiar with i.e. directions, procedures, etc.) every day both here at school and when you're at home - and so does KIBO! Explain to the students that if they want KIBO to do something, they need to give KIBO step by step instructions using the KIBO blocks.

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.

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)

Together as a class, students program their KIBOs to do the Hokey-Pokey. The teacher should show each individual block motion block and have KIBO react to it. Be very intentional in showing that each block represents an action for KIBO. Make the connection that just like how our instructions gave us directions one step at a time, we are creating a program one block at a time.

HOKEY-POKEY REFLECTION (5 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 (10 min)

On the Appendix B-1 you will find assessment A. Please hand out one copy of the assessment to each 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.

Lesson 7: 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

- Why is KIBO Confused? (20 min)
- Free Play (20 min)
- Debugging Reflection (20 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

MATERIALS

FOR THE TEACHER:

- Why is KIBO Confused? anchor chart
- Job Cards
- 1 flathead screwdriver
- Extra AA batteries
- Appendix E for troubleshooting tips

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

VOCABULARY

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

Lesson 7: Activities

WHY IS KIBO CONFUSED? (20 min)

In Lesson 6, students shared challenges of scanning KIBO blocks and other issues that they experienced while creating their Hokey-Pokey 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.

Have the students play out their solutions to better help them remember these solutions in the future.

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 (20 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 8: Cause and Effect

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 one new modules: the Sound Sensor. The Sound Sensor uses an event (wait for clap) before performing the subsequent action.

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

- Exploring Our Senses (10 min)
- KIBO Sound Sensor (5 min)
- Free Play (15 min)
- Reflection (10 min)
- Share Creations (10 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

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 *There Was an Old Lady Who Swallowed a Fly* by Simms Taback
- Job Cards
- 1 flathead screwdriver
- Extra AA batteries
- Appendix E for troubleshooting tips

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

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
- Sound — a type of energy made by vibrations in the air that we can hear

Lesson 8: Activities

EXPLORING OUR SENSES (10 min)

Ask students: What body parts do humans use to sense things in our environment? 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 say what senses the Old Lady is experiencing.

Taste	*What does the Old Lady taste?*
Smell	*What does the Old Lady smell?*
Touch	*What does the Old Lady feel?*
Hearing	*What does the Old Lady hear?*
Sight	*What does the Old Lady see?*

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 use events to trigger KIBO.

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.

FREE PLAY (15 min)

Individually or in pairs, students should take this time to explore the Sound Sensor module 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., sing block tells KIBO to sing). Encourage students to try other noises, like stomping or ringing a bell, to trigger the Sound Sensor!

REFLECTION (10 min)

Before sharing their free play creations, have students take out their Design Journals and use the KIBO stickers to write out their program.

SHARE CREATIONS (10 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.)? Why did you choose a particular block in your program? What was fun or challenging about creating their program? Did their program get KIBO to do what they wanted?*

SOLVE-IT ASSESSMENT B (10 min)

On the Appendix B-1 you will find assessment B. Please hand out one copy of the assessment to each 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.*

Lesson 9: Repeat Loops

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 (15 min)
- Free Play (15 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
- ☐ Print Design Journals (one for each student)*
- ☐ Print Toothbrush Sequence Pictures*
- ☐ 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:

- KIBO Says cards
- 1 copy of *There Was an Old Lady Who Swallowed a Fly* by Simms Taback
- Toothbrush Sequence Pictures*
- Job Cards
- 1 flathead screwdriver
- Extra AA batteries
- Appendix E for troubleshooting tips

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, Repeat and End Repeat blocks, number parameters

*See Appendix A for examples

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 9: Activities

REPETITION IN STORIES AND SONGS (15 min)

Throughout the book *There Was an Old Lady Who Swallowed a Fly*, certain phrases and words are repeated multiple times. Reread, if needed, and come up with a list of what phrases or words are repeated.
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 *There Was an Old Lady Who Swallowed a Fly*, 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)* Hand out Toothbrush Sequence Pictures to the children. Working in pairs, have students arrange the pictures in order, noting which pictures may be repeated. Then have the children come up with instructions for brushing your teeth based on the pictures they arranged and 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 (15 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.

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.

FREE PLAY (15 min)

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.

Lesson 10: Final Project - Characterization

Powerful Idea From Computer Science:
Representation

Powerful Idea From Literacy:
Character, Point of View

OVERVIEW

Students will begin their final project in this lesson. It is a three part final project that involves characterization, retelling and expansion. 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. In this lesson, students will learn about characters in literacy and learn how to characterize KIBO.

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 working in steps and creating 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

- There Was an Old Lady Who Swallowed a Fly (10 min)
- Design A Character (10 min)
- Create Your Own Character (20 min)
- Character Share (10 min)
- Solve-It Assessment C (10 min)

STUDENTS WILL BE ABLE TO...

- Design their own characters and reflect on the creation process
- Re-create their characters in 3-D to attach them to KIBO

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:

- 1 copy of *There Was an Old Lady Who Swallowed a Fly* by Simms Taback
- Job Cards
- 1 flathead screwdriver
- Extra AA batteries
- Appendix E for troubleshooting tips

FOR STUDENTS:

- Design Journal (see Appendix C for example)
- Art and craft materials
- KIBO bodies, wheels, motors, Begin and End blocks, blue Motion blocks, yellow Light blocks, Beep and Sing blocks, Wait for Clap blocks, Sound sensors, Repeat and End Repeat blocks, number parameters, platform pieces

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

THERE WAS AN OLD LADY WHO SWALLOWED A FLY (10 min)

Read the book *There Was an Old Lady Who Swallowed a Fly* by Simms Taback as a class. If needed, read the book a second time. Lead a student-centered discussion that reviews the story. Introduce the word characters, if this is a new vocabulary word for your students, spend time teaching the definition. A character is someone whose activities, thoughts and feelings are revealed in a story. It does not always have to be a person. The reader learns about the characters through details in the story.

If they are already familiar with the concept of characters, spend time in discussion.

Ask students: Who are the main characters in There Was an Old Lady Who Swallowed a Fly? Describe the characters in There Was an Old Lady Who Swallowed a Fly.

DESIGN A CHARACTER (10 min)

In their Design Journals, have the students create their own character based on The Old Lady or their imagination. You may want students to draw in pencil first and then go back and use crayons or other craft materials.

CREATE YOUR OWN CHARACTER (20 min)

Have the students create 3-D versions of the characters they designed using any art materials found in the classroom. These characters will be attached to KIBO's fixed or rotating art stage platforms to help characterize KIBO as their character.

CHARACTER SHARE (10 min)

Have students sit in a circle to share their characters. Encourage students to verbalize their thinking and reasoning behind their characters. For example, *ask students: Describe your character. What are they like? What materials did you make your character out of and why? What was easy about creating your character? What was difficult? Does your character look like what you wanted it to look like? In the future, how can you change it?*

SOLVE-IT ASSESSMENT C (10 min)

On the Appendix B-3 you will find assessment C. Please hand out one copy of the assessment to each 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.*

Lesson 11: Final Project - Retelling

Powerful Idea From Computer Science:
Representation, Algorithms, Sequence

Powerful Idea From Literacy:
Character, Point of View, Summarizing/Retelling the Sequence of a Story

OVERVIEW

In this lesson, students build upon their character creations from the previous lesson by programming their KIBO to reenact scenes from *There Was an Old Lady Who Swallowed a Fly* by Simms Taback. All of the elements students have learned will come together when students retell *There Was an Old Lady Who Swallowed a Fly* to each other.

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 working in steps and creating 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

- Reread (10 min)
- Sequencing Animals (5 min)
- Begin Coding the Retell (30 min)
- Share Projects (15 min)

STUDENTS WILL BE ABLE TO...

- Program KIBO to “swallow” animals in the order that the old lady did
- Devise their own way to show that KIBO is “swallowing” each animal
- Reflect on the decisions they made and difficulties they encountered in programming KIBO

PREPARATION FOR TEACHERS

- ☐ Read through the Activity Guide
- ☐ Print pictures for Sequencing Animals activity*
- ☐ 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:

- Animal Sequence Pictures*
- Job Cards
- 1 flathead screwdriver
- Extra AA batteries
- Appendix E for troubleshooting tips

FOR STUDENTS:

- Design Journal (see Appendix C for example)
- Pictures of animals from *There Was an Old Lady Who Swallowed a Fly**
- 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, Repeat and End Repeat blocks, number parameters, platform pieces

*See Appendix A for examples

Lesson 11: Activities

REREAD (10 min)

The purpose of this activity to have students think creatively about how they can bring *There Was an Old Lady Who Swallowed a Fly* to life, through KIBO. If necessary, reread the story to students.

Give the students the opportunity to learn about retelling stories by having them recount *There Was an Old Lady Who Swallowed a Fly* to their friends and classmates.

Ask students: Imagine (insert student name) was sick one day and missed There Was an Old Lady Who Swallowed a Fly, how would you tell them what happened in the story?

SEQUENCING ANIMALS (5 min)

Provide the images of animals from *There Was an Old Lady Who Swallowed a Fly* to each group. Using their sequence skills, have the students place the animals in the order the Old Lady swallowed them.

BEGIN CODING THE RETELL (30 min)

Before students begin programming KIBO to retell the story of *There Was an Old Lady Who Swallowed a Fly*, have the students identify the beginning, middle and end. After reminding the students the importance of a beginning, middle and end, take out KIBOs and blocks. Explain to students that today, KIBO will be the Old Lady from There Was an Old Lady Who Swallowed a Fly and we will be retelling the story with KIBO, but the students will get to decide for themselves what KIBO does to show that she/he is swallowing an animal.

Ask students: What do you think your robot should do to show that she/he is swallowing an animal (e.g., shake once it is over the animal, flash lights, sing a particular song)?

Students should program their robots to move over each animal in the order they were swallowed in the book, but when their KIBO is over each animal, students can choose whichever actions they would like to indicate swallowing the animal. Encourage them to get creative with their programs and challenge themselves through making program plans that will use advanced parts such as sensors, and repeat loops. Encourage students to keep track of their progress using their Design Journals.

SHARE PROJECTS (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: How did you decide how many motion blocks to use to get to each animal? What were the different ways you tried to trigger the Sound sensor (clapping, talking, etc.)? Why did you choose a particular block in your program?*

Additionally, have students practice giving helpful feedback without hurting feelings. Have children share with each other what they enjoyed and what they think could be improved.

- Give examples of helpful feedback such as: “I like how you….”
- Discuss various problems students faced and how they fixed them.
- Discuss what makes a clear retell.
- Do any students have tips to help their classmates solve problems they faced?
- What would they work on if they had more time?

Lesson 12: Final Project - Expansion

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

OVERVIEW

In this final lesson, students will expand upon their programs from the previous lesson to envision what could have happened if the Old Lady from *There Was an Old Lady Who Swallowed a Fly* by Simms Taback had done things differently. 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. By working in steps and creating 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

- Treading Along a Familiar Path (10 min)
- Design an Expansion (10 min)
- Program the Story (20 min)
- Share Creations (10 min)
- Solve-It Assessment (10 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
- ☐ 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:

- Job Cards
- 1 copy of *There Was an Old Lady Who Swallowed a Fly* by Simms Taback
- 1 flathead screwdriver
- Extra AA batteries
- Appendix E for troubleshooting tips

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, Repeat and End Repeat blocks, number parameters, platform pieces

Lesson 12: Activities

TREADING ALONG A FAMILIAR PATH (10 min)

The purpose of this activity to have students think creatively about what could have happened in *There Was an Old Lady Who Swallowed a Fly* if the Old Lady had done things differently.

Reread the book as a class, but this time, add more to the story than what is actually written. Read the story a second time, and this time let the children add to the story. Introduce the word elaborate to the children and really emphasize the fun that can be had with expanding upon a story. Elaborating is the process of adding or developing more details to an idea, a thought, a story, etc. than was actually presented. Elaborating does not necessarily require following the pattern of the story; it can involve changing the entire story and keeping just the same characters or changing everything but still being inspired by the original story.

Ask students: How else can we add onto this story? Do we include more animals? Fewer? What would have happened if the Old Lady had not swallowed the fly? If the fly had flown back out? If the Old Lady had not died at the end?

Have students think about these hypothetical scenarios and have several students share out their hypotheses.

DESIGN AN EXPANSION (10 min)

Now the students have the opportunity to turn these suggested scenarios into new, alternate stories. Students will draw their alternative stories in their Design Journals. This activity is also an opportunity to review whatever skills the students have learned about literacy. Distribute blank pages for students to draw new characters on.

Encourage students to let their imaginations go wild, but also stress that they have to have a beginning, a middle, and an end to their stories. Tell the students that they can use the elements that they have learned in novel ways to go along with their stories (e.g. shaking when over an image of an animal could represent petting rather than eating an animal).

PROGRAM THE STORY (20 min)

Bring out the KIBOs and the blocks and distribute them accordingly. Have each group or child tell you how they have elaborated on There Was an Old Lady Who Swallowed a Fly and how they plan to use the elements in their stories. Both let and encourage the children to be as creative and as thoughtful as they would like to be with their stories. As KIBO moves along, have the children tell you their stories.

SHARE CREATIONS (10 min)

During the final presentations, have students present their 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, how they decided which actions would mean certain things
- 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.

SOLVE-IT ASSESSMENT D (10 min)

On the Appendix B-4 you will find assessment D. Please hand out one copy of the assessment to each 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.

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)

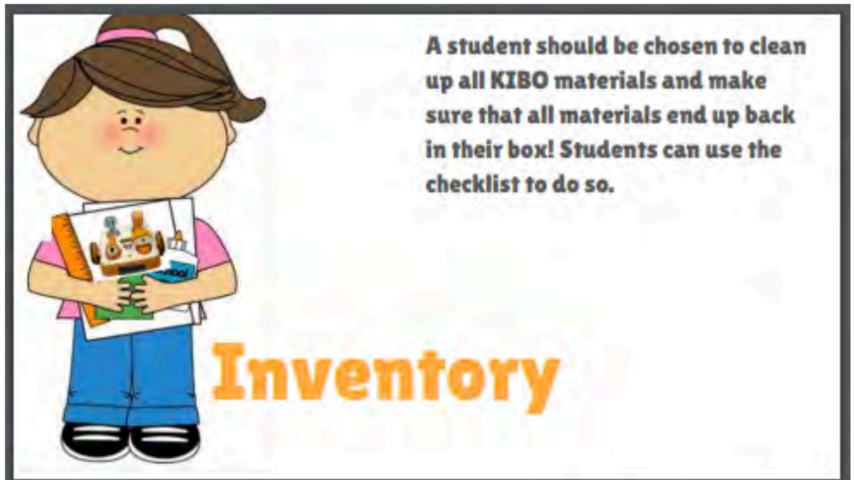
Teaching Materials:

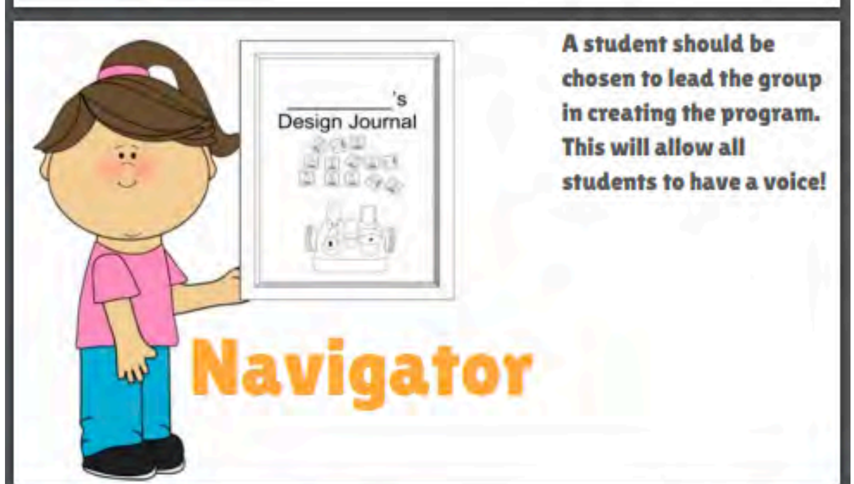
- ❒ 1 copy of *There Was an Old Lady Who Swallowed a Fly* by Simms Taback
- ❒ Large KIBO Says cards (purchase from KinderLab Robotics or make your own)
- ❒ Job Cards (**see following pages for examples**)
- ❒ Premade anchor charts **(see following pages for examples)**
    - ❒ How to Treat Our Materials
    - ❒ 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
    - ❒ Cow
    - ❒ Simms Taback
    - ❒ Toothbrush Sequence Pictures
    - ❒ Animal Sequence Pictures

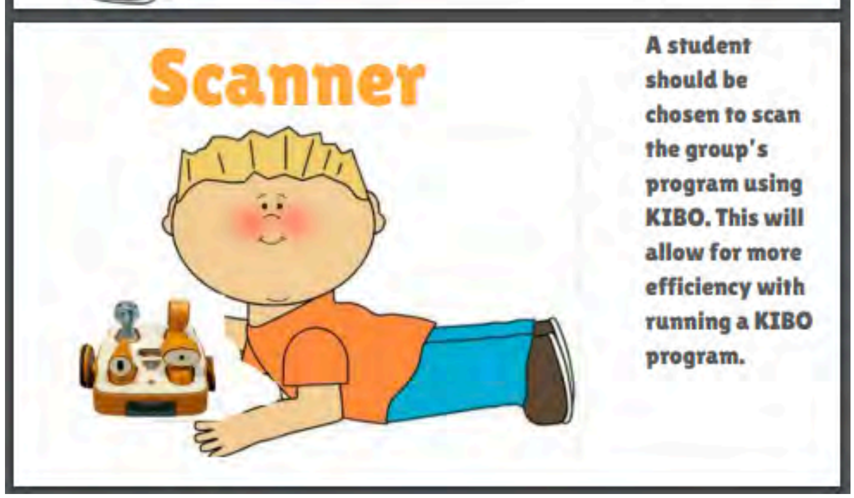
Optional Materials:

- ❒ 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 Job Cards:



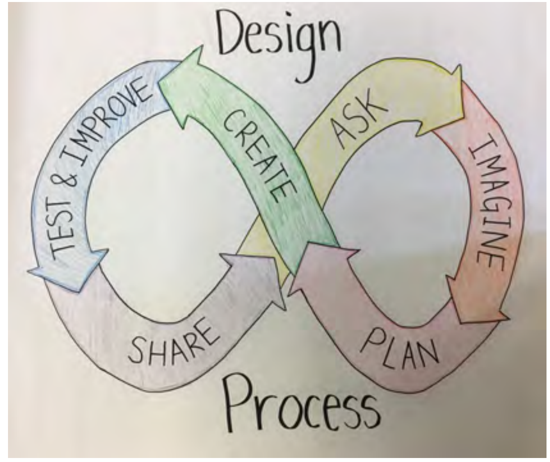




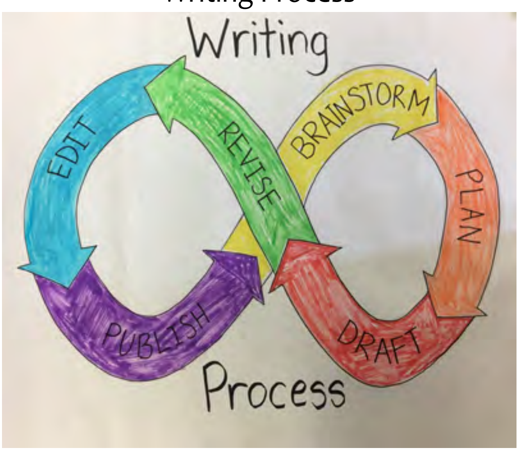
Examples of Anchor Charts:

How to Treat Our Materials

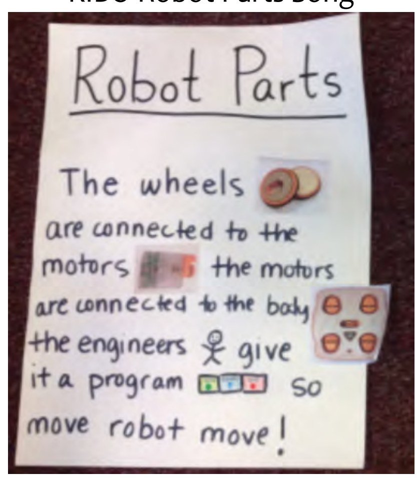
Design Process



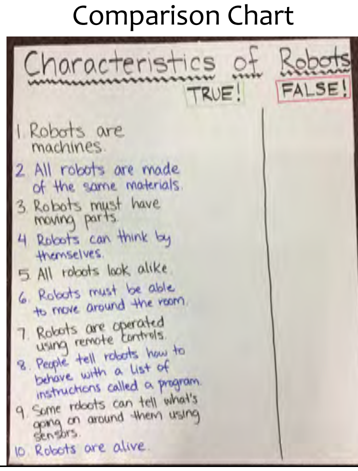
Writing Process



KIBO Robot Parts Song



Characteristics of Robots Comparison Chart



Why is KIBO Confused?

Scanning tips

To scan, hold KIBO 2–4" away from the bar code. Shine the red scanner light onto the bar code. It's ok if the light is a little "bigger" than the bar code.

If KIBO won't scan, try changing KIBO's position slightly. Move it slightly closer or farther away from the block and try changing the angle a little bit.

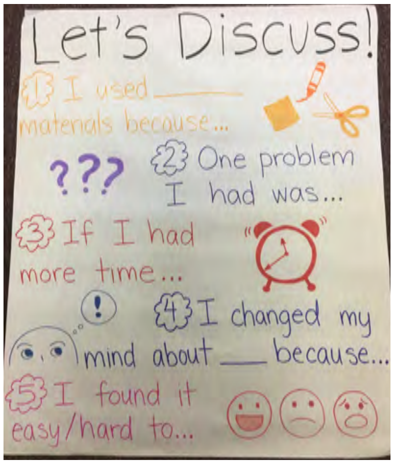
If you're still having trouble scanning, notice if there is light reflecting from your stickers. Try moving away from direct overhead lighting and windows. Or, try scanning the sides of the blocks, instead of the top.

This type of reflection can interfere with scanning.

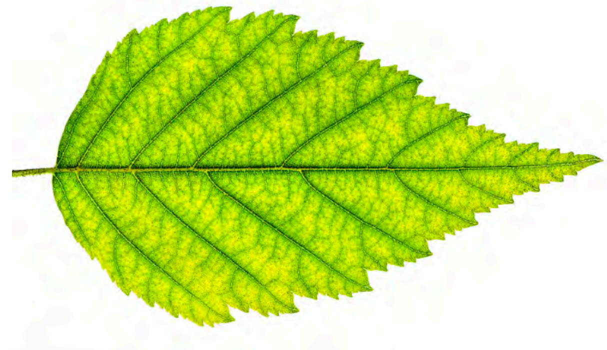
If scanning from the top isn't working ...

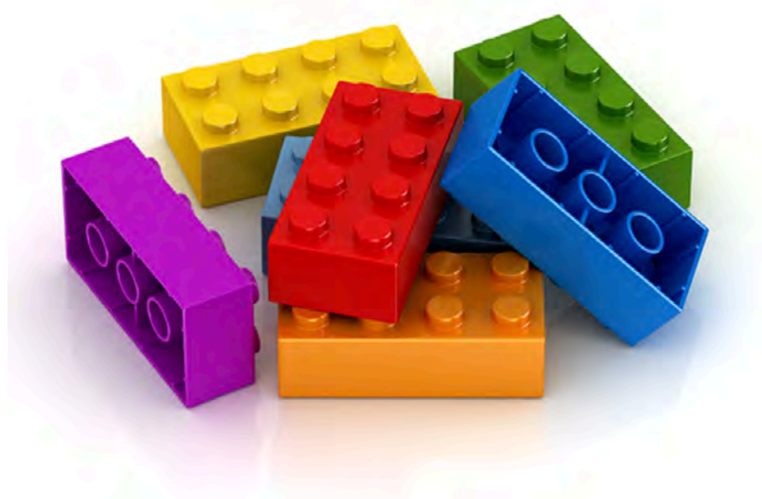
Try scanning from the side!

Examples of Anchor Charts:

Sentence Starters


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





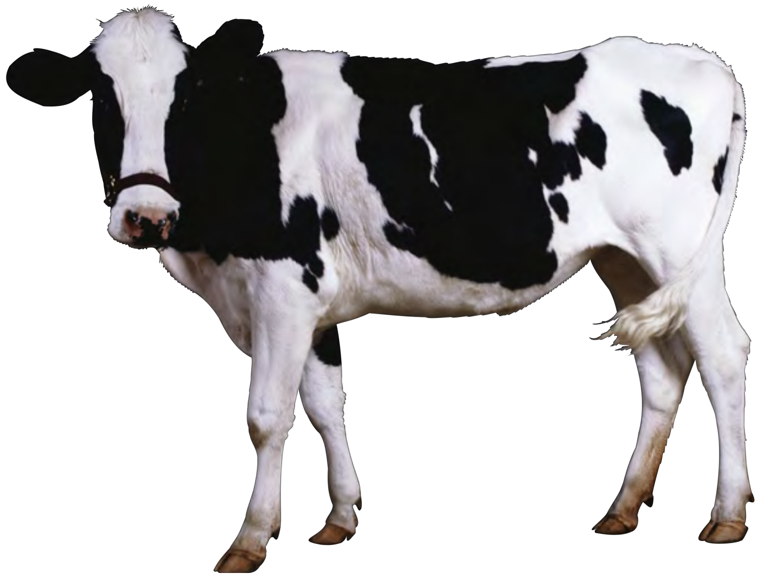
Examples of Printed Pictures:

8-10 pictures of robots and non-robots

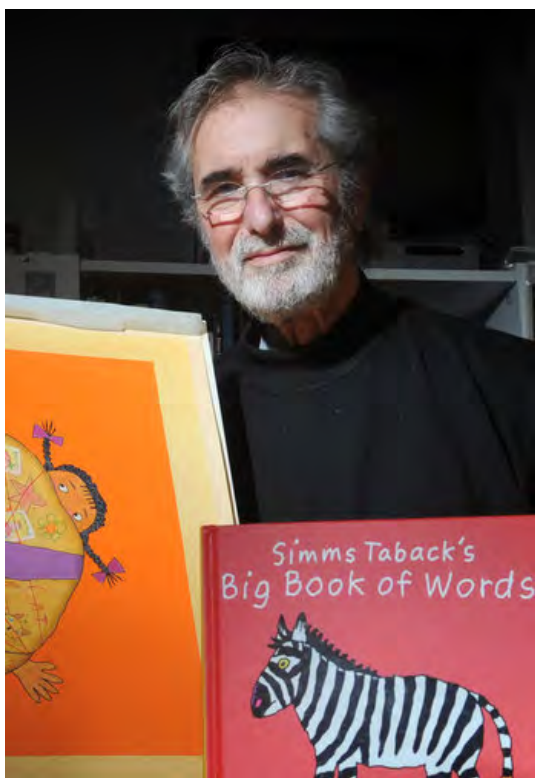
For non-robot pictures: reuse the images from the man-made objects in Lesson 1

Examples of Printed Pictures:

Cow



Simms Taback



Examples of Printed Pictures:
Toothbrush Sequence Pictures

turn on tap 	take toothbrush 	apply toothpaste 
rinse brush 	brush top 	brush bottom 
brush side 	brush side 	spit in sink 
rinse mouth 	rinse brush 	wipe face 

Examples of Printed Pictures:
Animal Sequence Pictures

Appendix B-1. Solve-It Assessment A

Solve-it
A

Name ______________________ Date _____________

Teacher ____________________

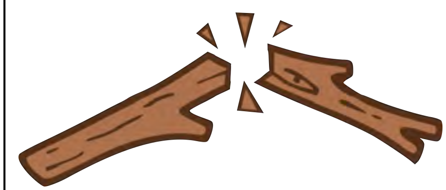
Circle the correct answers.

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

- Keeping things the same

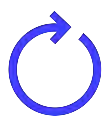
- Making new things

- Breaking something

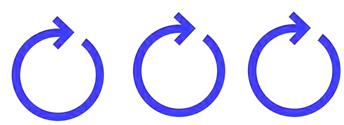


2. How many times can you go through the writing process and design process?

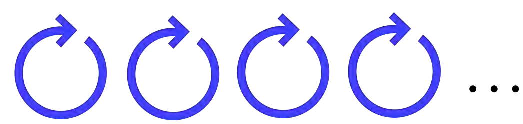
- 1 time



- 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

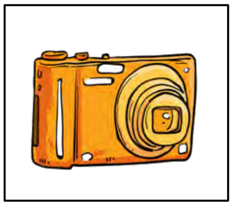
- A KIBO and a bike

- A KIBO and a house

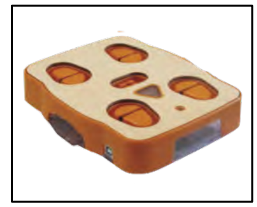
Name ____________________________ Date _________________

Teacher _________________________

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



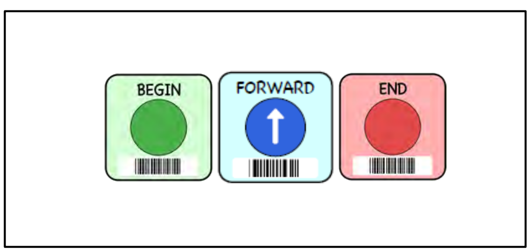


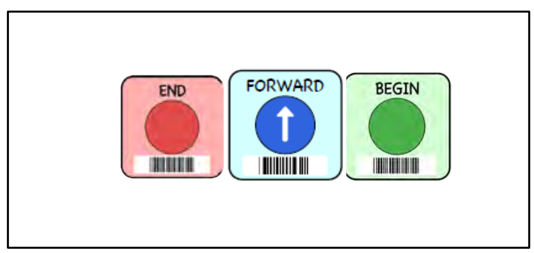
5. Which one of these is a robot?

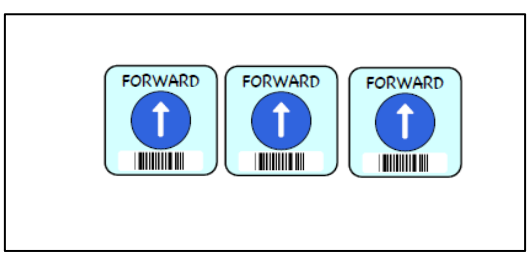


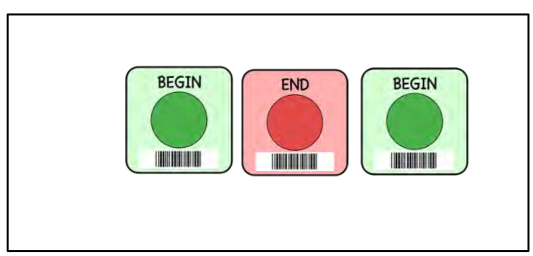


6. Which one of these will make a program that moves?









Appendix B-2. Solve-It Assessment B

Solve-it

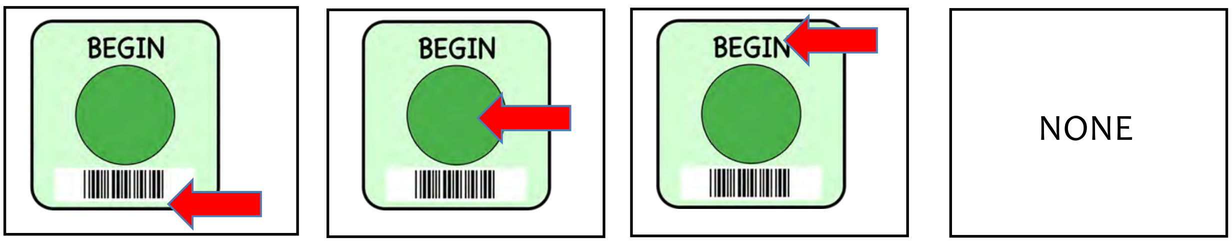
B

Name ____________________________ Date _________________

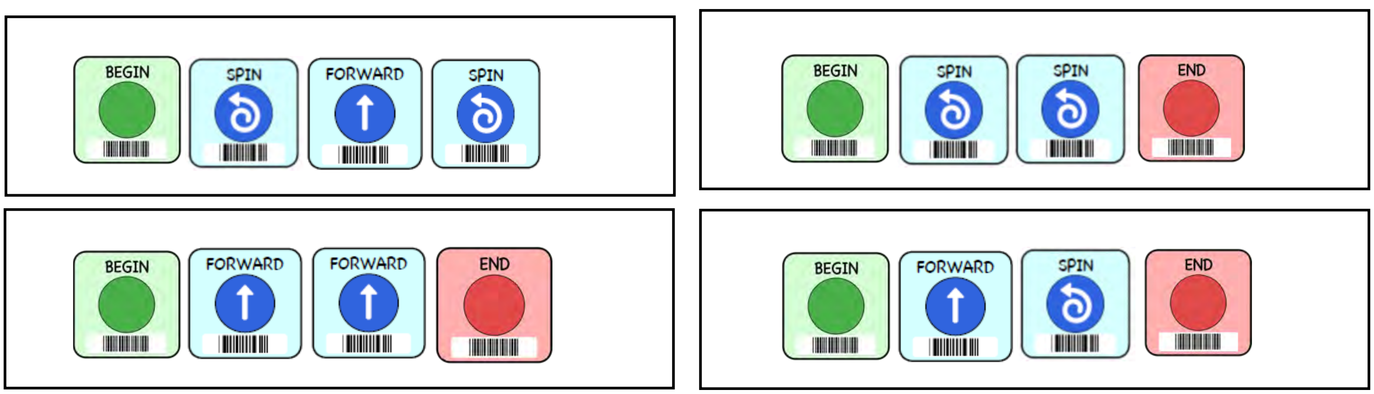
Teacher ____________________________

Circle the correct answers.

1. How does KIBO know what instructions to follow?



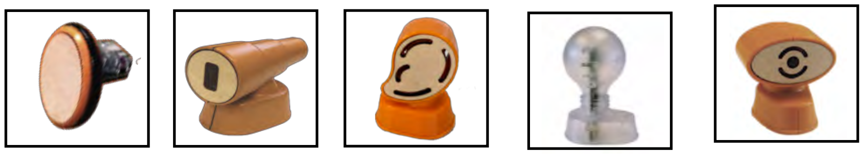
2. How can we change this program so that KIBO spins 2 times?



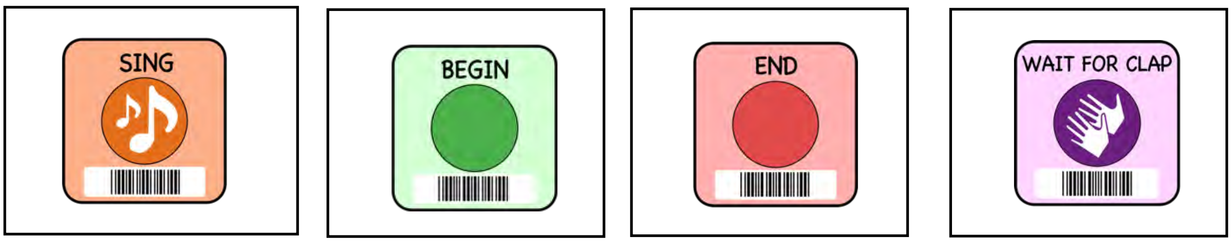
3. KIBO won't move forward, turn, or beep after I scan this program

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

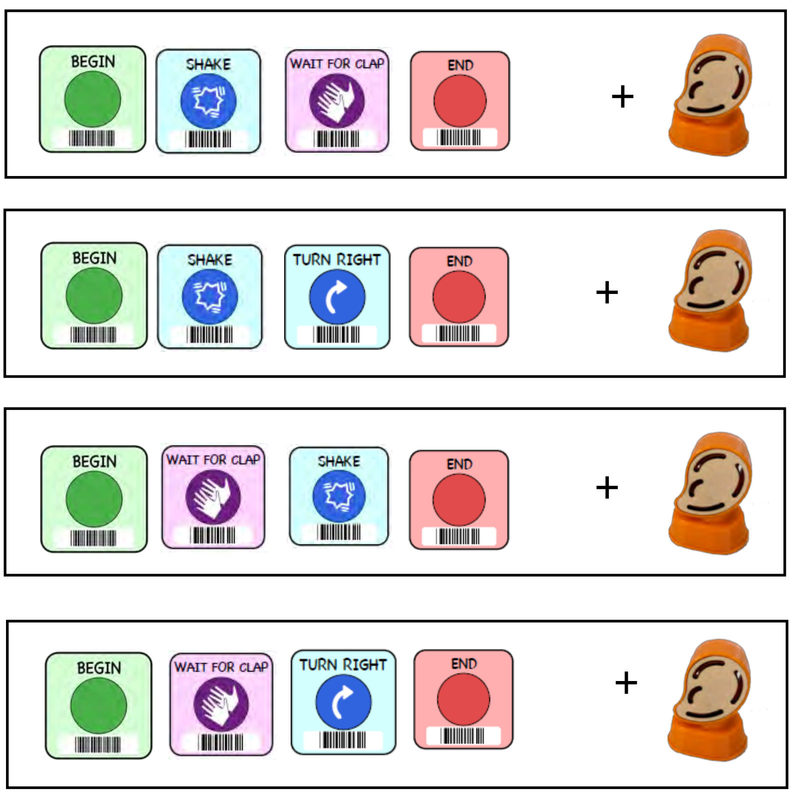
4. Circle the one of these that allows KIBO to listen.



5. Which block makes the program stop and start only after it hears a clap?



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



Appendix B-3. Solve-It Assessment C

Name ______________________ Date ______________

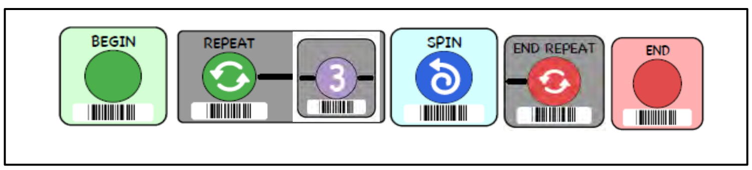
Teacher ____________________

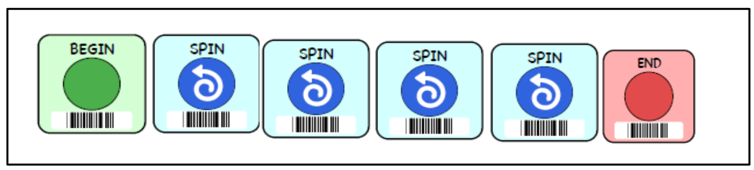
Circle the correct answer.

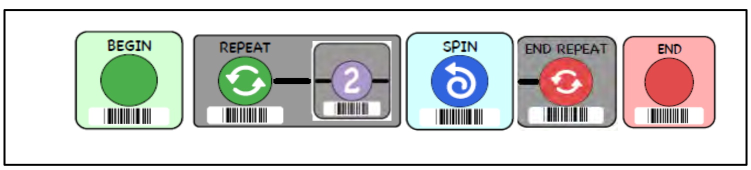
1. How many times will this program put a red light on?

1 2 3 4

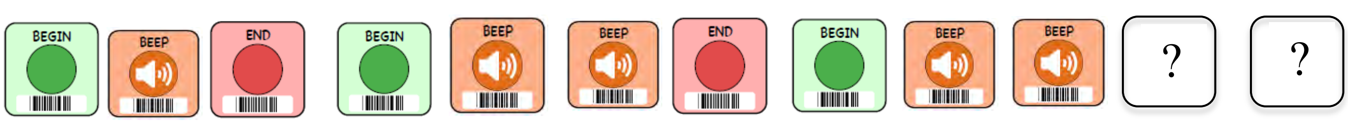
2. Which of these programs will make KIBO spin exactly 3 times?

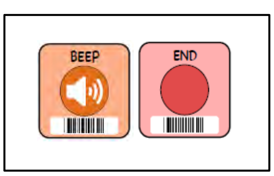


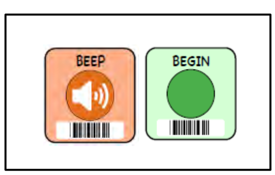


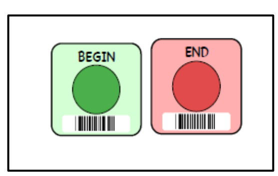


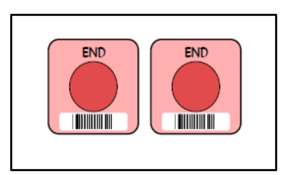
3. Which pair of blocks would complete this pattern?











Solve-it

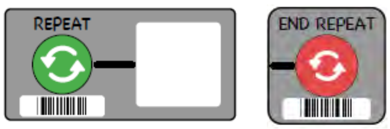
C

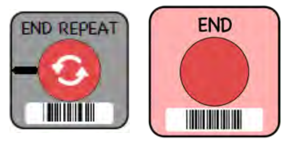
Name ______________________ Date ______________

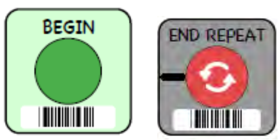
Teacher ______________________

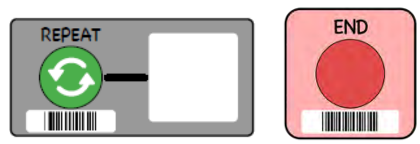
Circle the correct answers.

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









5. True or False: This program repeats the spin forever.

TRUE

FALSE

6. I want KIBO to spin three times, but my program isn't making KIBO spin three times:

How can I change my program to have KIBO spin three times?

Appendix B-4. Solve-It Assessment D

Solve-it

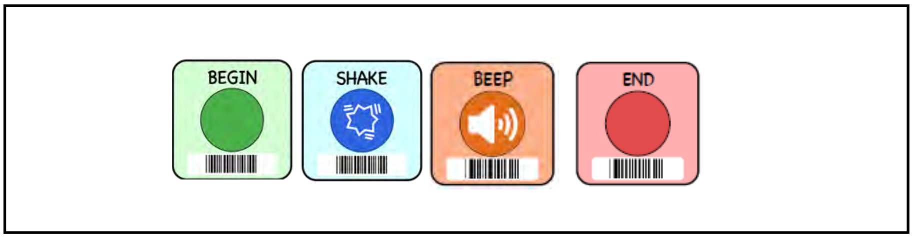
D

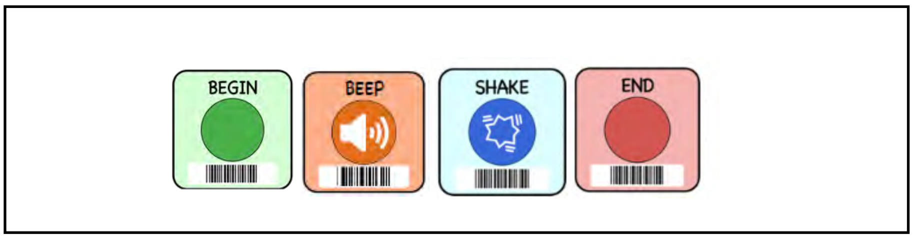
Name ____________________________ Date _________________

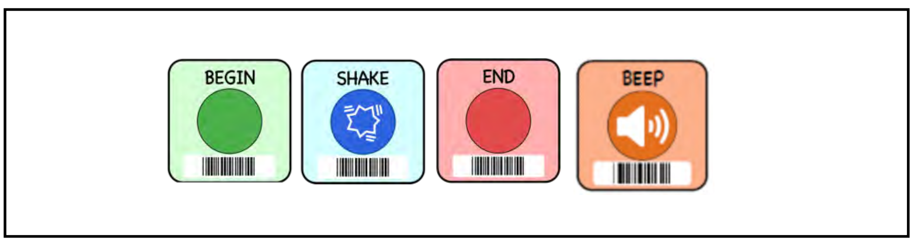
Teacher ____________________________

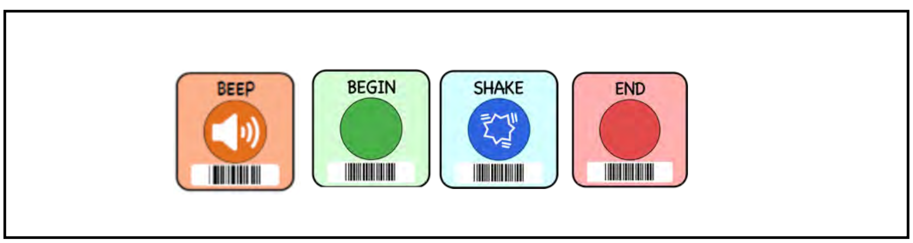
Circle the correct answers.

1. Which program will make KIBO beep and then shake?



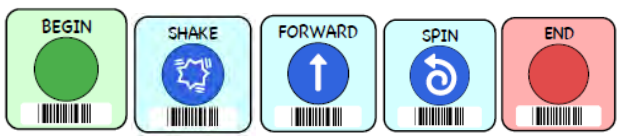




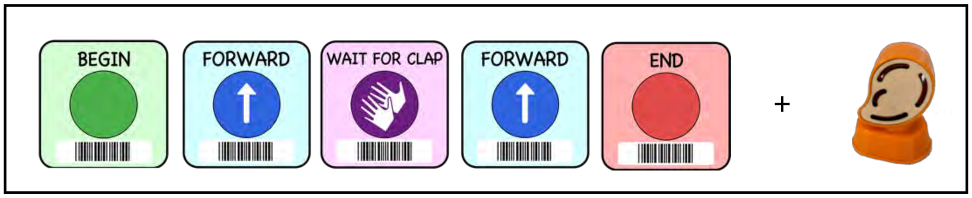


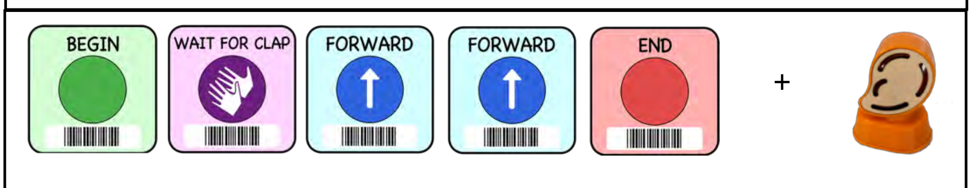
2. I want KIBO to shake, move back and spin. What is wrong with my program?

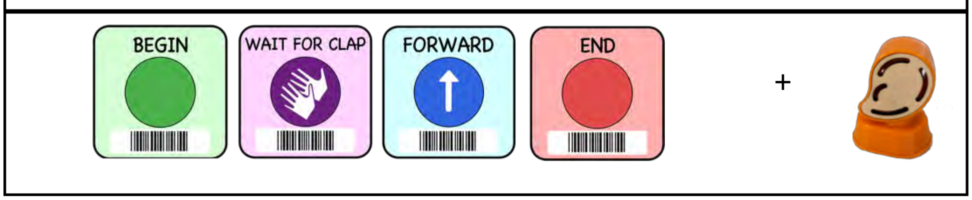
Circle the answer that will fix my program so that KIBO moves the way I want it to.



3. KIBO goes forward but waits until it hears a clap before continuing to go forward. Which program makes KIBO go forward, but wait until it hears a clap before going forward again?

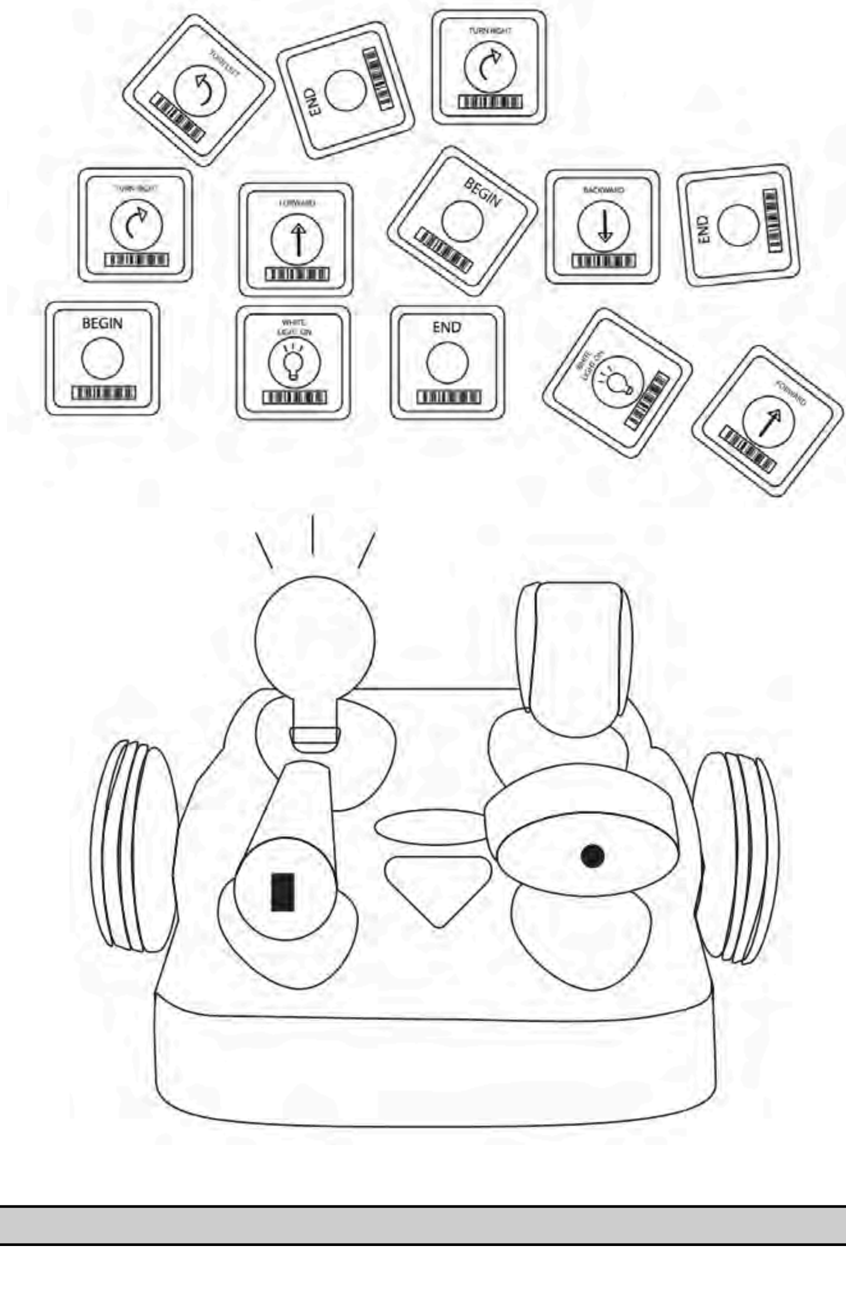






Appendix C. Design Journal

____________’s Design Journal



Name: ____________________ Date: _____________________

Lesson 1:
How to Build a Robot

Draw your robot here.

Name: ____________________ Date: _____________________

Lesson 1:
How to Build a Robot (continued)

What kinds of shapes are in your robot?

Draw the parts of a robot here.

Name: ____________________ Date: _____________________

Lesson 1:
How to Build a Robot (continued)

How do the parts go together?

What is the difference between a person and a robot?

Name: ____________________ Date: _____________________

Lesson 1:

Use this blank sheet for any extra planning!

Name: ____________________ Date: _____________________

Lesson 2:

Use this blank sheet for any extra planning!

Name: ____________________ Date: _____________________

Lesson 3:

Use this blank sheet for any extra planning!

Name: ____________________ Date: _____________________

Lesson 4:

Use this blank sheet for any extra planning!

Name: ____________________ Date: _____________________

Lesson 5:

Use this blank sheet for any extra planning!

Name: ____________________ Date: _____________________

Lesson 6:
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 6:

Use this blank sheet for any extra planning!

Name: ____________________ Date: _____________________

Lesson 7:

Debugging Reflection

What was one problem you had with KIBO?

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

Name: ____________________ Date: _____________________

Lesson 7:

Debugging Reflection (continued)

Which solution worked best?

Name: ____________________ Date: ____________________

Lesson 7:

Use this blank sheet for any extra planning!

Name: ____________________ Date: ____________________

Lesson 8:

Cause and Effect

Draw what you want your robot to do.

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

Name: ____________________ Date: _____________________

Lesson 8:

Cause and Effect (continued)

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

Name: ____________________ Date: _____________________

Lesson 8:

Use this blank sheet for any extra planning!

Name: ____________________ Date: _____________________

Lesson 9:

Use this blank sheet for any extra planning!

Name: ____________________ Date: _____________________

Lesson 10:

Design a Character

Draw your own character that you will program here.

Name: ____________________ Date: _____________________

Lesson 10:

Use this blank sheet for any extra planning!

Name: ____________________ Date: _____________________

Lesson 11:
Final Project

Now that you have drawn your character, show what you will make your character do!

Ask: What activities do you want your robot to do?

Imagine: What is your robot like?

My robot is a(an) ______________________________________

Its name is ____

Name: ____________________ Date: _____________________

Lesson 11:
Final Project (continued)

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:

Name: ____________________ Date: _____________________

Lesson 11:

Final Project (continued)

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

Draw the materials you used to decorate your KIBO:

Name: ____________________ Date: _____________________

Lesson 11:

Final Project (continued)

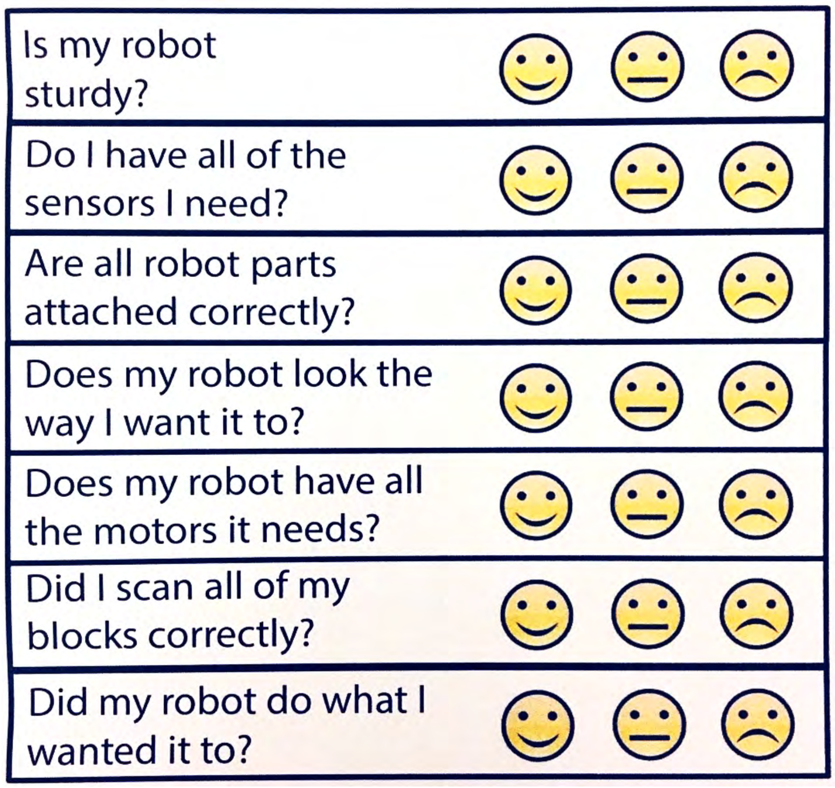
Use the KIBO stickers to write your robot's first program here. Don't worry if your first program doesn't work the way you want it to. You can always go back and make changes!

Name: ____________________ Date: _____________________

Lesson 11:

Final Project (continued)

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”!

Name: ____________________ Date: _____________________

Lesson 11:

Final Project (continued)

What changes did you make to your project?

Name: ____________________ Date: ____________________

Lesson 11:

Final Project (continued)

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: _____________________

Lesson 11:

Use this blank sheet for any extra planning!

Name: ____________________ Date: _____________________

Lesson 12:

Use this blank sheet for any extra planning!

Name: ____________________ Date: _____________________

Student Perceptions Survey

1. **How much did you like doing these activities?**

Not at all! | Not really | It was okay | Liked it | Loved it!

2. **How much do you think the whole class liked doing these activities?**

Not at all! | Not really | It was okay | Liked it | Loved it!

3. **How much did you learn about coding?**

Nothing! | A little bit | A fair amount | Quite a bit | Learned a lot!

4. **How important do you think the things you learned are going to be to you later in life?**

Not at all important! | Probably not | Maybe | Probably | Very important!

5. **How would you feel if you had the chance to do this again?**

Awful | Not very good | Okay | Pretty good | Great!

6. **What was your favorite part about these activities?**

____

____

Appendix D. Kinder Lab's Troubleshooting Tips

Getting started with KIBO

1. If this is your first time using your KIBO, insert 4 AA batteries into the battery case. The red scanner light will start blinking.

Screwdriver and batteries not included

2. Choose the motors, wheels, and sensors that you want to use. Insert the motors so that the green dot shows through KIBO's transparent bottom.

3. Sequence some blocks into a program. Every program needs a BEGIN block and an END block.

4. Push KIBO's triangular button to turn KIBO on. The red scanner light will blink.

KIBO will turn itself off if left alone for a few minutes.

5. Use KIBO to scan the bar codes on the programming blocks, left to right, one at a time*. If your scan was successful, KIBO will beep and the scan indicator LED will glow green after each block.

(A red scan indicator LED indicates a scanning error.)

**See scanning tips on the back of this guide.*

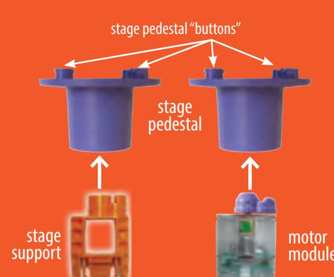
6. Push KIBO's triangular button to tell KIBO to go!

To re-run the program: push KIBO's button again.

To change your program: re-arrange the blocks, re-scan, and push KIBO's button. Watch KIBO go!

7. Decorate KIBO with the round or rectangular stage.

Insert stage support or motor module into stage pedestal



Insert the wood stage onto the stage pedestal buttons

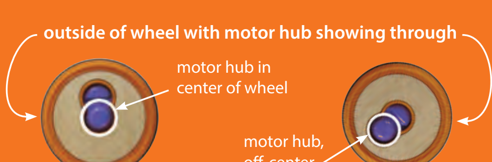
Insert stage assembly into middle motor socket

Decorate and play!

Fun things to try

Try inserting the motors "upside-down," with the green dot **not** showing and see what happens.

Insert the motors into the wheels so that the motors' axles are ***off-center***, relative to the center of the wheels, and see what happens!



Check out more fun challenges and activity guides at **http://resources.kinderlabrobotics.com.**

Good things to know

KIBO's lights can tell you lots of useful things:

◦ KIBO's red scanner light and triangular button will blink when KIBO is ready to ***scan*** a program – OR – when KIBO is ready to ***run*** a program. The button will stop blinking while KIBO is scanning a program; the red scanner light will stop blinking while KIBO is running a program.

◦ When KIBO's triangular button blinks, it means that KIBO has a program stored in its memory. The triangular button will go dark while KIBO is scanning a new program, and also after inserting new batteries.

◦ You can put KIBO to sleep by pressing and holding the triangular button for several seconds.

KinderLab robotics

Tips & Troubleshooting

Uh-oh ...

If the red scanner light is not blinking, it usually indicates a problem with the batteries. Remove and re-install the batteries. If that doesn't help, replace the batteries with new ones.

A tri-tone sound and a red scan LED means that an error occurred. KIBO may have mis-scanned, or there may be an error in your program. Try scanning again or re-arranging your blocks. Have fun experimenting!

If KIBO is turning left or right (or going backward) when it should be going forward, check the motors to make sure that the green dots are showing through KIBO's transparent bottom.

Programming tips

Make sure you plug in the sensors that your program needs! If you use the WAIT FOR CLAP block, you will need the "ear" (sound) sensor. If you use the LIGHT or DARK parameter cards, you will need the "eye" (light) sensor. If you use the NEAR or FAR parameter cards, you will need the "telescope" (distance sensor). If you use the RED/WHITE/BLUE LIGHT ON blocks, you will need the light bulb.

Take care of your motor modules!

KIBO's motor modules are designed to turn KIBO's wheels; they are not designed to carry a lot of weight. So, please don't force KIBO to go faster than it wants, and don't push down on KIBO's body when its wheels and motors are installed. These behaviors can damage the motor modules. Our warrantee doesn't cover damage caused by improper motor use.

Scanning tips

To scan, hold KIBO 2–4" away from the bar code. Shine the red scanner light onto the bar code. It's ok if the light is a little "bigger" than the bar code.

If KIBO won't scan, try changing KIBO's position slightly. Move it slightly closer or farther away from the block and try changing the angle a little bit.


This type of reflection can interfere with scanning.

If you're still having trouble scanning, notice if there is light reflecting from your stickers. Try moving away from direct overhead lighting and windows. Or, try scanning the sides of the blocks, instead of the top.

If scanning from the top isn't working ...

Try scanning from the side!

Join our KIBO community!
Sign up for our email newsletter for KIBO news, activity ideas, classroom tips, and more, at
http://kinderlabrobotics.com

Visit us on Facebook **http://facebook.com/KinderLabRobotics**

Follow us on Twitter: **@KinderLabRobot**

Colors of some components may vary.
Additional parts available at **shop.kinderlabrobotics.com**
Complete parts list at **kinderlabrobotics.com/compare**

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.