ROBOTIC ANIMALS

Kindergarten

A Curriculum Unit on Programming and Robotics

Integrated with Foundational Biology Topics

DevTech Research Group

Eliot Pearson Department of Child Development

Tufts University

http://ase.tufts.edu/DevTech/tangiblek/

Robotic Animals by DevTech Research Group is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License.

Under this license, you may use and adapt this work but you must attribute the work to the DevTech Research Group. You may not use or adapt this work for commercial purposes.

1 This curriculum was developed by Amanda Puerto and Amanda Sullivan with the help of Louise Flannery, Elizabeth Kazakoff, Adriana Flores, and Ethan Peritz, under the direction of Prof. Marina U Bers.
# Table of Contents

Introduction to the Curriculum ........................................................................................................4

Materials ........................................................................................................................................6

Pedagogy .........................................................................................................................................7

Classroom Management ..................................................................................................................12

Managing Materials .........................................................................................................................14

Student’s Assessments .....................................................................................................................15

Academic Frameworks Addressed: ................................................................................................16

Lesson 1 – What is the Engineering Design Process? ..................................................................29

Lesson 2 – What is a Robot? ..........................................................................................................33

Lesson 3 – What is Programming? ................................................................................................37

Lesson 4 – What are Repeats? ........................................................................................................40

Lesson 5 – What are Sensors? .........................................................................................................43

Lesson 6 – What are Ifs? ..................................................................................................................46

Lesson 7 – Final Project ..................................................................................................................49

Appendix A  Robotics across Themes ............................................................................................51

Appendix B  Songs and Games ........................................................................................................53

Appendix C  The Engineering Design Process ..............................................................................55

Appendix D  Design Journals ..........................................................................................................64

Appendix E  A Sample Engineer’s License .....................................................................................84

Appendix F  Positive Technological Development ..........................................................................88

Appendix G  Working with CHERP and the LEGO® RCX .............................................................91

Appendix H  Building Instructions ..................................................................................................93

Appendix I  Materials .......................................................................................................................95

Appendix J  Assessments ..................................................................................................................101

References .....................................................................................................................................110
Tables and Figures

Tables

Table 1: Powerful Ideas within the Activities.................................................................21
Table 2: Integrating Biology.......................................................................................... 24
Table 3: ITEEA Standards Addressed..........................................................................26

Figures

Figures 1-7: The Engineering Design Process.............................................................. 57
Figure 8: Sample Engineer’s License..........................................................................85
Figure 9: Personal Development Trajectory...............................................................89
Figure 10: Collaboration Web.....................................................................................89
Figure 11: Possible Mobile Robot Designs.................................................................94
Introduction to the Curriculum

This curriculum introduces powerful ideas from computer science, specifically programming in a robotics context, to 1st and 2nd grade children in a structured, developmentally appropriate way. While the curriculum uses the technology of the LEGO® Mindstorms™ robotics construction set, the powerful ideas are applicable to any other robotic construction kit. The term powerful idea refers to a central concept within a domain that is at once personally useful, 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: the engineering design process, robotics, programming, repeat parameters, and sensors. These powerful ideas are explored in the context of a curriculum that draws on the theme of animals and can be adapted to many other early childhood themes (see Appendix A). Each unit follows the same basic structure: 1) warm up games to playfully introduce or reinforce concepts, 2) introduction of the powerful idea through a challenge, 3) work individually or in pairs, 4) technology circle, 5) free-explorations, and 6) student’s assessment. Teachers should adapt the lesson structure and its components to suit their class’s needs.

Since many modern graphical user interfaces are not designed with the developmental needs of such young learners in mind, they are generally ill-suited for use in early elementary school classrooms, especially for computer programming activities. To overcome this problem, this research project has created a tangible-graphical hybrid programming language specifically for young children, the Creative Hybrid Environment for Robotics Programming, or CHERP. See http://ase.tufts.edu/DevTech/tangiblek/research/cherp.asp for more information.

Rather than using a keyboard to type programs to control robots, children using CHERP physically construct programs by connecting interlocking wooden blocks with labels which both a computer and young child can recognize. Children also have the option to use graphical icons manipulated on-screen with a mouse, or to switch between the two interfaces. This hybrid approach
creates a unique opportunity to separate the intellectual act of computer programming from the confounding factors of many programming interfaces. It therefore provides a medium for young learners to experience success with computer programming of robotic objects.

Just as young children can read age appropriate books, computer programming can be made accessible by providing young children appropriate tools. When implemented with a curriculum such as the following, CHERP provides a powerful tool for young children to program with.

**Pacing**

The curriculum unit is designed to take place over the course of one intensive week of work (i.e. in a camp setting or during a robotics focused week at school) or over the course of 1-2 months with one or two shorter sessions per week. These numbers are certainly not set in stone. Depending on children’s developmental levels and prior experience with digital technology, programming, and robotics, students might need more or less time than the guidelines here indicate. One issue for each teacher to resolve is how long to allot for each session, keeping in mind that each lesson can be spread out over several sessions to accommodate the classroom schedule and students’ attention spans for this work. Depending on the students, a class may benefit from between 1 and 2 hours to devote to their robotics and programming activities at a time.

Some classes or students may benefit from further division of the activities into smaller steps or from more time to explore each new concept before moving onto the next, either in the context of free-exploration or with teacher-design challenges. Each of the powerful ideas here can easily be expanded into a unit of study; the activities provide an introduction to each concept. For instance, students could explore a range of different activities and challenges with sensors to learn how they work in more depth.

To supplement the structured challenges, two to three hours of free-exploration are allotted throughout the curriculum. These open-ended sessions are vital for children to fully understand the
complex ideas going on with 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. Rather, if time is tight, teachers can consider leaving out a particular lesson altogether, giving children enough time to really understand and work with the ideas they are introducing to rather than skimming over all the lessons presented in this curriculum. Free-exploration provides opportunities for playing with materials and ideas. This will help build a solid foundation.

**Materials**

The robotic pieces referred to in this curriculum come from the LEGO® Company’s Mindstorms™ robotics construction kit. In the next few years there may be other robotics construction sets on the market for early childhood education and which could be used with this curriculum. A second, and important, type of material used in the curriculum is the incorporation of inexpensive crafts and recycled materials. Current robotic construction kits utilize materials such as LEGO®s®, which can be very expensive and challenging for young children with small hands to use. They also do not necessarily appeal to all children and require some creativity to incorporate with other kinds of 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. It may also bring down the costs associated with acquiring robotic supplies, since a blend of materials may reduce the need for complete, LEGO®-based kits and instead require the purchase of only select robotic parts which give functionality and movement to the creations.

For example, a kindergarten class in Boston created a robotic Freedom Trail, using cardboard boxes to recreate the historical buildings of the city. The children integrated the use of relatively more
expensive LEGO® pieces, such as light sensors, to bring their constructions to life². This activity, while engaging for the children, also proved very successful for the teacher, who was already familiar with the use of recyclable materials and felt less intimidated by not having to learn about the mechanics of LEGO® bricks.

**Pedagogy**

The theory of constructionism developed by Seymour Papert (Papert, 1980; 1993) shows that children learn best when they construct digital artifacts and knowledge by playing with and exploring concrete materials. The social context of these explorations is also crucial, and teachers can provide scaffolding by creating a learning environment that supports children’s explorations and experimentation. Through questions and observations, the teacher engages students in articulating and extending their own observations, thought processes, and explorations. The teacher may not directly answer students’ questions but rather show them how to find it themselves. This kind of exploration fosters an environment in which what we often see as “failure” is actually a natural step of the learning process, a signal to ask questions and explore further. A more detailed account of working with young children and technology, especially robots, can be found in *Blocks to Robotics: Learning with Technology in the Early Childhood Classroom* (Bers, 2008). See that book or an excerpt from it included in Appendix D for a description of supporting students with planning versus tinkering styles of approaching robotics, programming, designing, and problem-solving.

**Theoretical Framework: Positive Technological Development**

The theoretical foundation of this robotics curriculum is called Positive Technological Development (PTD) and was developed by Prof. Marina U. Bers from Tufts University (Bers, 2010a; Bers, in press). 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, 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 robotics. These are: content creation, creativity, communication, collaboration, community building and choices of conduct.

This curriculum engages young learners in:

1. **Content creation**, by engaging children in making a robotic artifact and in 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 make transparent to the children themselves, as well as teachers and parents, their own thinking, their learning trajectories and the project’s evolution over time. The design process, like the scientific method, gives students a tool for systematically addressing a problem or need. Some children need constraints and top-down planning. Others enjoy working bottom-up and messing around with the materials to come up with ideas. Both learning styles are conducive for building competence in the technological domain and are welcomed and respected in this robotics program by providing different kinds of design journals explicitly designed to encourage differentiated instruction. See Appendix D for an explanation of how design journals should be used and for a sample design journal.
2. **Creativity**, by integrating different media such as LEGO® pieces, motors, sensors, recyclable materials, arts and crafts, and graphical elements from the programming language. This curriculum promotes creativity by fostering opportunities for children to develop their own project ideas, as opposed to copy them from a booklet or follow instructions, and by providing different materials for children to work with. As children approach solving technical problems in creative ways, they develop a sense of confidence in their learning potential. The PTD approach of working with robotics is based on the promotion of creativity, as opposed to efficiency, in problem solving. The original meaning of the term **engineering**, which derives from the Latin *ingenium*, means “innate quality, mental power, clever invention”. However, clever or creative projects are difficult to make and the process can be frustrating. Our approach aims at helping children to learn how to manage frustration. One way for children to manage frustration in difficult lessons is by taking occasional breaks from the task that is frustrating them. For example, if a child is getting increasingly agitated working on a programming task, it is okay for them to take a few minutes to step away and work on the aesthetics of their robot and return to programming when they feel ready. Teachers can also schedule in “breaks” during long building and programming periods if the class is growing frustrated and bring everyone together to reinforce concepts and troubleshoot problems. The learning environment is set up to create a culture in which it is expected for things not to work and in which succeeding the first time is seen as a rarity and as a sign that the child might not have challenged him or herself. Learning how to manage frustration is also associated with the development of confidence. As children go through this program they slowly realize their ability to find solutions, either by trying multiple times, by using different strategies or by asking for help. This is an important aspect of emotional development.

3. **Collaboration**, by engaging children in a learning environment that promotes working in teams, sharing resources and caring about each other. Most educational robotic programs
for older children, such as the National Robotics Challenge and FIRST (For Inspiration and Recognition of Science and Technology), are set up as competitions, events where robots have to accomplish a given task – usually out-perform another robot. However, this program, instead of focusing on competition, promotes caring and working together. For that purposes it utilizes the collaboration web, a tool used to foster collaboration and support. At the beginning of each day of work, each child receives, along with their design journal, a personalized printout with his or her photograph in the center of the page and the photographs and names of all other children in the class arranged in a circle surrounding that central photo (see Appendix F for an example). Throughout the day, at the teacher’s prompting, each child draws a line from his or her own photo to the photos of the children with whom he or she has collaborated. 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. At the end of the week, children write or draw “thank you cards” to the children with whom they have collaborated the most.

4. **Communication**, through mechanisms that promote a sense of connection between peers or with adults. One of the ways this program engages students in communication is through technology circles - a time for all, children and adults, to stop their work, put their projects on the table or floor, sit down in a circle together, and share the state of their projects. Technology circles present a good opportunity for problem solving as a community. Technology circles can be called as often as every twenty minutes at the beginning of a project, or only once at the end of a day of work, depending on the needs of the children and the need of the teacher to introduce new concepts. Some teachers have all the children sit together in the rug area for this. There are challenges to be addressed in order for technology circles to successfully serve their purposes. Kids’ excitement to use the materials during introductory conversations or their post work tiredness during wrap-up discussions may necessitate special attention to the structure and expectations for group discussions.
For instance, these conversations may need to be rather concise, making it a challenge to cover the many complex ideas presented in this curriculum’s activities. To cover all the important ideas without losing the children’s attention, the discussions might be broken up and held throughout the day rather than all at once. 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 help them 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.

5. **Community-building**, through scaffolded opportunities to form a learning community that promotes contribution of ideas. The long term goal of this robotics program is not only to foster computational thinking and technological fluency amongst the participating children and teachers, but also amongst the wider community. In the spirit of the Reggio Emilia approach started by the Municipal Infant-toddler Centers and Preschools of Reggio Emilia in Italy after World War II, projects done by children are shared with the community via an open house, demo day, or exhibition (Rinaldi, 1998). These open houses provide authentic opportunities for children to share and celebrate the process and tangible products of their learning with others who are also invested in their learning, such as family, friends and community members. Public displays of the learning process serve a dual function: to make learning visible to others and to the children themselves. In this spirit, final projects are shown to parents, friends and community members in the form of open houses. During these open houses, parents, siblings, and other family members visit the class for a demonstration of the children’s final projects. 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.
6. **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. In every classroom, there is always a child who learns quickly about mechanics, and thus has the choice to help her classmates in need or to build a bigger structure. The same is true for those children who become programming experts, those who can problem-solve and those who can mediate conflicts amongst group members. Although differentiation of roles is important for growing a responsible learning community, children are also encouraged to take on new roles and be flexible. Choices of conduct are not only made by children. Teachers make important choices in the way they display and introduce the materials to the children. For example, if the LEGO® building pieces are sorted by types and placed in bins in the center of the room (instead of giving an already sorted robotic kit to each child or group), children learn how to take what they need without depleting the bins of the “most wanted” pieces, such as special sensors or the colorful LEGO® mini-figures. They also learn how to negotiate for what they need. 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. One way to encourage positive choices is by using “Expert Badges”. Children who master concepts quickly can earn Expert Badges (a sticker for them to wear). A child wearing an expert badge uses the remainder of the class period helping any students who are having difficulty with the concepts they have mastered. Children wearing Expert Badges and actively helping others will also have an easier time completing their collaboration webs. (See Appendix F for sample Expert Badges)

**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 classroom or 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.

**Group Sizes**

The curriculum refers to whole-group versus pair or individual work. In fact, some classrooms may benefit from other groupings. Piloting of this curriculum has shown that kindergarteners are better able to explore the main activities in the lessons when they have their own materials to work with and can go to other students for help, rather than collaborating with the same materials. 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, preferably individually, while working on the challenges. On the other hand, the curriculum includes numerous conversations which are enriched by multiple voices, viewpoints, and experiences. Some classes may be able to have these discussions as a whole group. Other classes may want to break up into smaller groups to allow more children the opportunity to speak and to maintain focus. Some classes structure robotics time to fit into a “center time” in the schedule, in which students rotate through small stations around the room with different activities at each location. This format gives students more access to teachers when they have questions and lets teachers tailor instruction and feedback as well as assess each students’ progress more easily than during whole-group work. It is important to find a structure and group size for each of the different activities (instruction, discussions, work on the challenges, and the final project) that meet the needs of the students and teachers in the class.
Managing Materials

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

The second question is of usability. In some cases, children’s desks or tables do not provide enough space to build a robot and program it on the computer. 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 or their interface preference.

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 make expectations for how to use and treat materials explicit. 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).
**Student’s Assessments**

Children employ many different concepts and skills to create and program their own robots. The assessments for each lesson distill those ideas down to the 2-3 core powerful ideas of each activity. They are scored on a scale indicating how successful children were at achieving that concept or skill, from 0 (cannot achieve) to 5 (completely achieves). The final projects employ most or all the concepts covered in the lessons, depending on students’ individual projects. See Appendix J for copies of assessments.

To keep assessment manageable in a busy classroom and also give children a tool to self-regulate their exploration process and self-assess, the assessment criteria given with each lesson can constitute a sequence of concrete achievements leading up to an “Engineer’s License.” Each lesson is associated with a different level, e.g. “Sturdy Builder” or “Programmer I,” that incrementally completes the license, at which point the child is ready to start a final project. During the course of each lesson, children will explore and learn at different rates. When they think they have accomplished the criteria for that lesson’s assessments, they demonstrate this to a teacher, who marks that licensure level on their certificate or helps them identify missing components. Children re-attempt any level until they have mastered it. This format allows for individual differences, helps teachers manage the amount of time assessment takes, and provides a fluid way for teachers to assess both individual progress and that of the whole class. Teachers should feel free to come up with their own analogy for the incremental assessment described here as “licenses.” One teacher likened it to levels of achievement in video games that you must complete before moving on to harder challenges. See Appendix F for a sample Engineer’s License.

The design journal for planning the final project can also provide a means of documentation. Appendix E shows a sample design journal. The writing can be done by children or by teachers taking dictation as appropriate. The components of the journal should be tailored for the nature of the final project. This format can also be adapted simply to document the final version of the project.
Academic Frameworks Addressed:

Kindergarten

This curriculum is designed for Kindergarten students and covers many foundational computer science and engineering skills that are not often taught in early childhood. These academic frameworks are taught through a series of powerful ideas: the Engineering Design Process, Robotics, Programming, Sensors, and Ifs. Each powerful idea has activities and materials (in this case, the activities are tailored to fit the animal theme) that encourage mastery of the powerful idea and the foundational academic subjects that support it. In addition, the curriculum addresses foundational math, literacy, science, and art skills. Within each lesson in this curriculum, there are descriptions of at least one math and one language arts activity that fit in with the powerful idea being taught. Below is a brief summary of how the academic frameworks fit into each of these powerful ideas.

Powerful Idea 1: The Engineering Design Process

The Engineering Design Process refers to a cyclical process engineers use to design an artifact to meet a need. Its steps include: identifying a problem, looking for ideas, developing a prototype, testing, improving, and sharing solutions with others. The Engineering Design Process is introduced in Lesson 1 and is used as a problem solving technique throughout the remaining lessons. The Engineering Design Process is a tool students can continually apply to all subjects, experiments, and projects in the future.

In Lesson 1 (What is the Engineering Design Process?) children use LEGO® and art materials to build something they see at the park. They are given various constraints in the design of their structures and must use the steps of the Engineering Design Process to plan, create, test, and share their structures until they have a prototype that meets all the constraints.

Math

Using LEGO® to build their structures, children must be able to identify and describe shapes as well as analyze and compare shapes (Common Core Standards, Grade K). Children will look at real
animals and keep a list of all the shapes they see. In their Engineering Design Journals (booklets filled with worksheets and blank space where children plan, answer questions, draw, etc.) children will complete accompanying activities regarding labeling the shapes they use.

**Language Arts**

In their Engineering Design Journals, children will use a combination of drawing, dictating, and writing to explain what they have designed and why (Common Core, Grade K). Children will recall information from the activity to answer questions in their journals (Common Core, Grades K-2). Children will participate in collaborative conversations with a partner or group (Common Core, Grades K-2). Finally, children share their work in the Technology Circle for the first time. Technology Circles are a daily activity when the class comes together to share their work, discuss what they have done that day, troubleshoot un-resolved questions as a group, and answer any questions. With prompting and support, children recount their experiences with appropriate facts and details in an articulate manner (Common Core, Grades K-2).

**Art**

In Lesson 1, children are asked to think creatively and artistically. They will draw out their structures during the “planning” stage in their Engineering Design Journals. After their LEGO® structures are built, they will use a variety of arts and crafts materials to decorate and their vehicles. This exploration of art materials and exercise in imagination fits in with the Benchmark 2nd grade Art Making Standards and it is important that children are exposed to these materials as early as Pre-K and Kindergarten.

**Powerful Idea 2: Robotics**

Robotics is an engineering field focused on the creation of robots, machines which can automatically follow instructions to do tasks. Robotics is first introduced in Lesson 2 (What is Robotics?).
In Lesson 2, children share and learn ideas about what robots are. They are introduced to basic robotics concepts. Using different robotics, LEGO®, and art materials children build functioning robotic animals.

**Math**

Children continue their work in describing shapes, and reasoning with shapes in the design and building of their robotic animals (Common Core Standards, Grades K-1).

**Language Arts**

In Lesson 2, children are introduced to a variety of new words and concepts when shown robotic materials for the first time. Through a variety of group and individual activities, children receive prompting and support for asking and answering questions about unknown words (Common Core Standards, grade K). Children continue to use their Engineering Design Journals for writing and drawing explanations for their work and answering questions (Common Core, Grades K-2). Children practice speaking skills in the Technology Circle (Common Core, Grades K-2).

**Science**

In this lesson, children use the Engineering Design Process to make predictions, make decisions, observe, and manipulate materials (Inquiry Skills, based on NYC MST Standards, Grades K-8).

**Powerful Ideas 3 and 4: Programming and Programming with Repeats**

A program is a sequence of instructions that the robot acts out in order. Each instruction as a specific meaning, and the order affects the robot's overall actions. Instructions can be modified with a special instruction to repeat. Parameters, extra pieces of information, can make loops repeat forever or for a specific number of times.

Programming is addressed primarily in Lesson 3 through the activities *What is Programming?* and *What are Repeats?* In Lesson 3, children first program their robots to do the Hokey Pokey. When this is
mastered, children use repeats correctly to make their robots move between two destinations on opposite ends of “road” with a turn in it.

Math

When learning to use the repeat parameters in their programs, children must know the number names and the count sequence (Common Core, Grade K). When learning about repeats, children recall what they know about patterns. They must be able to recognize a repeating pattern in a program and create patterns themselves (TERC, Grade K).

Language Arts

Children continue to use their Engineering Design Journals to answer questions and write/dictate/draw about their work (Common Core, Grades K-2). Children participate in structured collaborative conversations with a partner or group in order to troubleshoot problems (Common Core, Grades K-2). Children practice speaking skills in the Technology Circle (Common Core, Grades K-2).

Science

Children must observe, analyze, and report observations (Process Skills, based on the NYS MST Standards). Based on these observations, they make and record their predictions for how they will reach their goal (Process Skills, based on the NYS MST Standards).

Powerful Idea 5: Sensors

A robot can use sensors, akin to human sense organs, to gather information from its environment. Sensor data can become parameters for controlling flow instructions. Light sensors are introduced in Lesson 5 (What are Sensors?). Children correctly attach and program the sensors to their robotic animals.
Math

In their Engineering Design Journals, children will record data regarding the number of times their animal needs to go forward in order to get through the tunnel (Common Core, Grade K).

Language Arts

They continue to practice speaking skills in the Technology Circle and writing and literacy skills in their Engineering Design Journals (Common Core, Grades K-2).

Science

Children practice observing and inferring (Inquiry Skills, based on NYS MST Standards, Grades K-8). They compare and contrast the touch and light sensor (Inquiry Skills, based on the NYS MST Standards). They observe cause and effect relationships (Process Skills, based on the NYS MST Standards).

Powerful Idea 6: Ifs

A branch instruction (or “if”) tells a robot to follow one set of instructions or another based on a sensor’s state. In Lesson 6 (What are Ifs?), children program their robot to carry out different instructions depending on the current state of a sensor.

Math

Children practice using if-commands during a game of Red Light/Green Light. In this game, children exercise their memory of number names and count sequence (Common Core, Grade K). For example, children may be told to jump 3 times if the light is green and one time if the light is red.
Language Arts

Children work with new vocabulary words to practice writing and dictating “If scenarios” (Common Core, Grades K-2). They will complete a worksheet explaining what they do if it Saturday versus what they do if it is Monday.

The Final Project

The powerful ideas introduced in lessons 1-6 culminate in a final project that draws upon all of the building, programming, math, science, literacy, and art skills children have learned.

Table 1: Powerful Ideas within the Activities

<table>
<thead>
<tr>
<th>Powerful Idea</th>
<th>Definition</th>
<th>Activity</th>
</tr>
</thead>
</table>
| Engineering design process     | A cyclical process engineers use to design an artifact to meet a need. Its steps include: identifying a problem, looking for ideas, developing a prototype, testing, improving, and sharing solutions with others. | **What is the Engineering Design Process?**
Children will use LEGO® and art materials to build a non-robotic vehicle or animal.

**Goals:**
- To gain experience using the Engineering Design Process in order to solve problems and complete a task
- To gain expertise in sturdy building |
<table>
<thead>
<tr>
<th>Powerful Idea</th>
<th>Definition</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robotics</td>
<td>An engineering field focused on the creation of robots, machines which can automatically follow instructions to do tasks.</td>
<td><strong>What Is a Robot?</strong>&lt;br&gt;Children share and learn ideas about what robots are. They are introduced to robotics concepts.&lt;br&gt;&lt;br&gt;&lt;strong&gt;Goals:&lt;/strong&gt;&lt;br&gt;- To gain expertise in the different robotics parts and use this expertise to build their own robotics creations.&lt;br&gt;- To be exposed to a brief introduction to programming instructions.</td>
</tr>
<tr>
<td>Programming: Control Flow by Sequencing and Instructions</td>
<td>A program is a sequence of instructions that the robot acts out in order. Each instruction has a specific meaning, and the order of the instructions affects the robot’s overall actions. Instructions can be modified with a special instruction to repeat. Parameters, extra pieces of information, can make loops repeat forever or a specific number of times.</td>
<td><strong>What is Programming?</strong>&lt;br&gt;Children build and program a robotic animal to do the Hokey Pokey.&lt;br&gt;&lt;br&gt;&lt;strong&gt;Goals:&lt;/strong&gt;&lt;br&gt;- To gain expertise in what a program is.&lt;br&gt;- To figure out and program the proper sequence of instructions in order to make their robot do the Hokey Pokey.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>What are Repeats?</strong>&lt;br&gt;Children are introduced to the concept of repeats.&lt;br&gt;&lt;br&gt;&lt;strong&gt;Goals:&lt;/strong&gt;&lt;br&gt;- To use repeats correctly to make their robots travel between two given points.</td>
</tr>
<tr>
<td>Powerful Idea</td>
<td>Definition</td>
<td>Activity</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>Sensors</td>
<td>A robot can use sensors, akin to human sense organs, to gather information from its environment. Sensor data can become parameters for controlling flow instructions.</td>
<td><strong>What are Sensors?</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Children are introduced to the light sensor, add it to their robots, and use it to program their robots to turn on their lights when it is dark.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Goals:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- To gain an understanding of sensors and how they work.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- To use a light sensor appropriately with their robots.</td>
</tr>
<tr>
<td>Control flow by instruction:</td>
<td>A branch instruction tells a robot to follow one set of instructions or another based on a sensor’s state.</td>
<td><strong>What are Ifs?</strong></td>
</tr>
<tr>
<td>Branches</td>
<td></td>
<td>Children program their robot to carry out different instructions depending on the current state of a sensor.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Goals:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- To understand the concept of if.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- To understand that a robot can ‘choose’ between two sequences of instructions depending on the state of a sensor.</td>
</tr>
</tbody>
</table>
Integrating Biology with Robotics & Programming

The Robotic Animals curriculum integrates foundational biology topics with powerful ideas from programming and robotics. While the Robotic Animals curriculum includes activities that focus on biology, the robotics piece can be modified to fit in with numerous other early childhood content areas (see Appendix A for examples). In addition to the math and language arts connections found in each lesson of this curriculum, you will also find a biology connection. The biology connections are derived to meet the Massachusetts DOE Frameworks for the Life Sciences. This curriculum contains activities that specifically address the following frameworks: differentiating between living and nonliving things, grouping living and nonliving things by characteristics they share, recognize that animals interact with their environment using their senses, and identifying the way an animal’s habitat provides for its basic needs. See the chart below for a detailed explanation of how biology is integrated into each lesson of the curriculum.

Table 2: Biology Connections in the Robotics Animals Curriculum

<table>
<thead>
<tr>
<th>Lesson</th>
<th>Biology Connection</th>
<th>Biology Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lesson 1: What is the Engineering Design Process?</td>
<td>• Characteristics of living and nonliving things</td>
<td>• Visit a park and keep a list of all the things they see. Later, children categorize these items into 2 lists: living and nonliving things.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Children will act as engineers and make nonliving representations of something they saw at the park.</td>
</tr>
<tr>
<td>Lesson 2: What is a Robot?</td>
<td>• Reinforce characteristics of living and nonliving things</td>
<td>• Compare and contrast groups of animals (e.g., insects, birds, fish, mammals) &amp; look at how animals in these groups are more similar to one another than to animals in other groups.</td>
</tr>
<tr>
<td></td>
<td>• Compare and contrast groups of animals</td>
<td>• Children pick an animal to build a robotic version of.</td>
</tr>
<tr>
<td>Lesson</td>
<td>Biology Connection</td>
<td>Biology Activity</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------</td>
<td>------------------</td>
</tr>
</tbody>
</table>
| Lesson 3: What is a Program? | • How do different animals move?  
• What are the physical characteristics of different types of animals? | • Children will think about the way different types of animals move (i.e. how fish move vs. how a dog moves) and dance the Hokey Pokey as various different animals. What unique body parts do some animals have that allow them to move differently (i.e. which animals have fins, tails, paws, etc.)?  
• Children will program their robot animal to dance the Hokey Pokey. Afterward, children will think about how they can modify their program so that their robot moves more the way their specific animal would move |
| Lesson 4: What are Repeats? | • Animal migration | • Children learn what animal migration is. As a class, think about the different types of animals that migrate.  
• When programming their own robotic animals to travel, children will be encouraged to think about where and why their animal might be traveling. |
| Lesson 5: What are Sensors? | • Human and animal senses | • If you have small animals in the classroom, observe them while they find food, water, shelter, etc. What senses are they using? Talk about how people and animals use their senses every day. |
| Lesson 6: What are Ifs? | • Animal sleep patterns- diurnal vs. nocturnal animals | • Children learn about diurnal vs. nocturnal animals and use this knowledge when programming their robotic animals to do one activity when it is day and another activity when it is night |
The International Technology and Engineering Educators Association (ITEEA) is a professional organization for technology, design, and engineering educators. The ITEEA promotes technological literacy by supporting the teaching of technology in schools. Additionally, the ITEEA strengthens the technology field through state and national legislative efforts, professional development, membership services, publications, and school activities. Please see Table 2 (below) to see how the ITEEA standards line up with the MA Science and Technology Frameworks and many of the powerful ideas found in this curriculum.

<table>
<thead>
<tr>
<th>Powerful Idea</th>
<th>International Technology and Engineering Educators Association Standards by standard and grade</th>
<th>MA Science and Technology / Engineering (STE) and Technology Literacy (TL) Frameworks by standard and grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engineering design process</td>
<td>- People plan to help get things done. (Std 2E; K-2)</td>
<td>- Engineering design requires creative thinking and consideration of a variety of ideas (and strategies) to solve practical problems (generated by needs and wants). (STE Std 2 Central Concept; PreK-2 (&amp; Gr 3-5))</td>
</tr>
<tr>
<td></td>
<td>- Everyone can design solutions to a problem. (Std 8A; K-2)</td>
<td>- Engineering design is an iterative process [...] (STE Std 2 Central Concept; Gr 6-8)</td>
</tr>
<tr>
<td></td>
<td>- Design is a creative process (that leads to useful products and systems). (Std 8B; K-2/Std 8C; Gr 3-5/Std 8E; Gr 6-8). All designs can be improved. (Std 8F; Gr 6-8)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- The engineering design process includes identifying a problem, looking for ideas, developing solutions, and sharing solutions with others. (Std 9A; K-2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Asking questions and making observations helps a person to figure out how things work. (Std 10A; K-2)</td>
<td></td>
</tr>
<tr>
<td>Powerful Idea</td>
<td>International Technology and Engineering Educators Association Standards by standard and grade</td>
<td>MA Science and Technology / Engineering (STE) and Technology Literacy (TL) Frameworks by standard and grade</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Robotics</td>
<td>-Build or construct an object using the design process. (Std 11B; K-2)</td>
<td>- With teacher direction, use appropriate technology tools [...] to define problems and propose hypotheses. (TL Std 3.6; Gr 3-5)</td>
</tr>
<tr>
<td></td>
<td>-Discover how things work. (Std 12A; K-2)</td>
<td>-Describe the various ways that objects can move, such as in a straight line, zigzag, back-and-forth, round-and-round, fast, and slow. (STE Physics Std 3; K-2)</td>
</tr>
<tr>
<td></td>
<td>-Systems have parts that work together to accomplish a goal (Std 2B; K-2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-Tools, machines, etc use energy to do work. (Std 16D; Gr 3-5)</td>
<td></td>
</tr>
<tr>
<td>Programming:</td>
<td>Recognize and use everyday symbols (Std 12C; K-2)</td>
<td>-Identify and explain how symbols and icons [...] are used to communicate a message (STE Tech Std 3.4; Gr 6-8)</td>
</tr>
<tr>
<td>Control Flow</td>
<td>-People use symbols when they communicate by technology (Std 17C; K-2)</td>
<td></td>
</tr>
<tr>
<td>by Sequencing</td>
<td>-The study of technology uses many of the same ideas and skills as other subjects. (Std 3A; K-2)</td>
<td></td>
</tr>
<tr>
<td>and Instructions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensors</td>
<td>-The natural world and human-made world are different. (Std 1A; K-2)</td>
<td>-Characteristics of natural and human-made materials (STE Tech Std 1.1; PreK-2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Human beings and animals use parts of the body as tools (STE Tech Std 2.2; PreK-2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Differentiate between living and nonliving things. Group both living and nonliving things according to the characteristics that they share. (STE Bio Std 2; K-2)</td>
</tr>
</tbody>
</table>
The Curriculum
Lesson 1
What is the Engineering Design Process?

Powerful Idea:
The Engineering Design Process

Overview:
Children use LEGO® and art materials (all non robotic materials) to build a model of something they see at the park. The powerful idea in Lesson 1 (building sturdily through use of the engineering design process) will prove important to the success of the children's robots in subsequent lessons and should be rearticulated and discussed during each activity.

<table>
<thead>
<tr>
<th>Prior Knowledge</th>
<th>Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Students will understand that...</td>
</tr>
<tr>
<td>• None, but prior experience building with LEGO® and crafts or recycled materials is helpful.</td>
<td>• LEGO® bricks and other materials can fit together to form <strong>sturdy structures</strong>.</td>
</tr>
<tr>
<td></td>
<td>• The <strong>engineering design process</strong> is useful for planning and guiding the creation of artifacts.</td>
</tr>
</tbody>
</table>

Materials / resources:
• LEGO® bricks and a variety of crafts and recycled materials for building and decorating
• Poster showing the steps of the engineering design process (see Appendix D)
• Engineering Design Journals for planning
• Pictures of different types of vehicles

Activity description

Warm-up: Fieldtrip to the Park
Visit a local park where children will be able to see various living and nonliving things. As a class, keep a list of all the different things you see (i.e. various animals, plants, benches, playground equipment, etc.) When you get back to the classroom, you will categorize these items as “living” and “nonliving”. Children will pick something in particular to record/draw in their Engineering Design Journals.

**Biology Connection:** Characteristics of Living Things
As a class, discuss ideas about the differences between living and nonliving things. (note: This lesson will become particularly important when children begin to learn about robots in subsequent lessons, as many children incorrectly believe robots are alive)

**LIVING THINGS:** anything that is alive or has ever been alive (e.g., a dog, a flower, etc.)

**NONLIVING THINGS:** anything that is not now and has never been alive (e.g., rock, mountain, glass, wristwatch)
**Math Connection:** Shapes at the Park

Ask children to identify different shapes they see making up the structure or animal they have chosen to record. Back in the classroom, look at LEGO® and art materials and ask children to identify the shapes they see and make another list of their answers. Compare the two lists. What shapes do they have in common? How can we use materials in the classroom to build structures that look like the real ones? In this activity, children will work to identify and describe 2d shapes as well as 3d shapes (see vocabulary section).

**Introduce the concepts and the task:** “Today we will be building something we saw at the park and we’re going to use a tool to help us make sure our structures are sturdy and work the way they are supposed to.” Discuss what an engineer is and introduce the steps of the engineering design process (see Appendix D for a poster).

**What is an engineer?**
An engineer is anyone who invents or improves things (for instance, just about any object you see around you) or processes (such as baking methods) to solve problems or meet needs. Any man-made object you encounter in your daily life was influenced by engineers.

**Think Like an Engineer**

Everyone in the class is going to start thinking like an engineer! That means looking at the purpose of objects and how they function. What are the different parts that make up the whole? What do they do? Why are they important? Let’s look at pictures and ask these engineer’s questions.

Ex 1: Fire engine- What are the different parts of the fire engine? What function does each part have? Why is each part important?

Ex.2: Ice Cream truck- What function does each part have? What parts are the same as the fire engine? What parts are different? Why?

**Jump For Engineers**

Look at a series of pictures of naturally occurring and manmade objects. Jump if you think an engineer built it; stay seated if you don't think so. Why or why not? Discuss. (See Materials in Appendix I for samples)

**Lesson 1 Vocabulary**

Students should become familiar with the following words:

- **Artifact** – something important made by people
- **Circle** – a round shape with no edges
- **Cycle** – something that moves in a circle (i.e. the seasons, the Engineering Design Process)
- **Design** – a plan for a building or invention
- **Edge** – the border of a shape
**Engineer** – someone who invents or improves things  
**Living** – anything that is alive or has ever been alive (e.g., a dog, a flower, etc.)  
**Material** – something used to build or construct  
**Nonliving** – anything that is not now and has never been alive (e.g., rock, mountain, glass, wristwatch)  
**Rectangle** – a shape with four sides, two pairs of sides with equal length  
**Square** – a shape with four equal sides  
**Triangle** – a shape with three sides

Individual / pair work: Students follow the steps of the engineering design process and use LEGO® and crafts or recycled materials to create a model of something they saw at the park. This could be an animal, a plant, a bench, a car, etc. They may use both structural and aesthetic materials. Students should demonstrate to a teacher that their structures meet the following criteria as they are ready.

The criteria for a successful structure are that:
- It is sturdy and remains intact when picked up or moved around
- It is designed to resemble something seen at the park

**Language Arts Connection: Postcard Home**

Children will recall their fieldtrip to the park. They will try to remember the things they saw and what they liked the most. Children will fill out a blank postcard (in Engineering Design Journals) where they will draw pictures describing their trip for them to send home to their families. With help, they can try to label their pictures using vocabulary words (or dictate the words for the teacher to label). When postcards are complete, cut them out so that children can mail them or take them home!

**Note: Working Individually vs. Working in Pairs**

Whether students work in pairs versus individually throughout this lesson is left up to the teachers’ discretion based on several factors. Materials may be limited, making pair work necessary. Teachers may also have goals for children’s social development that an explicit focus on sharing and teamwork throughout this curriculum can support. On the other hand, teamwork can be challenging at this age, so students may benefit from having their own materials and the option rather than the requirement to collaborate with others when it makes sense.

**Engineering Experts:** Children who finish building their vehicles and master all concepts quickly get to wear a badge that says “Engineering Expert”. Engineering Experts walk around and offer help to any classmates experiencing difficulties.

**Collaboration Web:** As children progress through the lesson, they will complete their collaboration webs. They will draw lines from their picture to the pictures of any classmates who give them help. If children say they didn’t receive any help, remind them to think of their partners, class Experts, or if they got any ideas by looking at another classmate’s project.
**Technology Circle:** After finishing, students share their creations. They may do one or more of the following:

- a. explain the features of their creation
- b. show how their creation moves (if applicable)
- c. describe the features of their final design that make it sturdy
- d. talk about what they found easy and difficult, and
- e. share anything they changed from their original plan.
- f. share their postcards
- g. share their collaboration webs

**Free-play:**

Provide opportunities for children to build freely with LEGO® and other arts and crafts materials.
Overview:
Children share and learn ideas about what robots are. They are introduced to robotics concepts. Children will build and test their own robotic animals.

<table>
<thead>
<tr>
<th>Prior Knowledge</th>
<th>Objectives</th>
<th>Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Students will understand that...</td>
<td>Students will be able to...</td>
<td></td>
</tr>
<tr>
<td>• LEGO® bricks and other materials can fit together to form sturdy structures.</td>
<td>• Robots need moving parts, such as motors, to be able to perform behaviors specified by a program.</td>
<td>• Describe the components of a robot, including the ‘brain,’ motors, and wires.</td>
</tr>
<tr>
<td>• The engineering design process is useful for planning and guiding the creation of artifacts.</td>
<td>• The robotic ‘brain’ has the programmed instructions that make the robot perform its behaviors.</td>
<td>• Upload a program to a robot</td>
</tr>
<tr>
<td>• Symbols (pictures, icons, words, etc) can represent ideas or things.</td>
<td>• The computer must communicate with the motors for the motors to function.</td>
<td>• Build a sturdy, robotic vehicle that moves.</td>
</tr>
<tr>
<td>• Some ability to recognize letters or to read is helpful, but not required.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Materials / resources:

• Pictures of different robots and non-robots
• Large icons for games and reference displays
• Computers with CHERP software
• One set of robotic parts for each student/pair
• LEGO® bricks and a variety of crafts and recycled materials for building and decorating
• Some partially built robots (or pictures of them) to show possible attachments

Note:
It is important to establish rules or expectations for how students should treat each others’ materials, programs, and robots. Find a time for students to generate these group expectations. Students may be better able to imagine reasonable expectations after using the robots or programming interface once.
Activity description

Warm up activities:

1) Jump for the robots! Children will be shown about 10 different images of robots and non-robots. They jump up and down if they think the picture shown is of a robot. Later, make an “Is It a Robot?” chart putting these images in one of three categories: Robots, Maybe or Sort of Robots, and Not Robots. (See Materials in Appendix I for sample pictures)

2) Yes or No? Students jump up (or make another movement) for statements they think are true and sit down for statements they think are false.

**Math Connection: Graphing Class Responses**

Incorporate graphing into this exercise by making a chart 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) with their initials in the “True” column or the “False” column. As a class, children will be able to interpret the graph in order to see whether there were more “True” or “False” responses for each question.

1. Robots are machines (YES). _____
2. All robots are made of the same materials (NO). _____
3. Robots must have moving parts (YES). _____
4. Robots can think by themselves (NO). _____
5. All robots look alike (NO). _____
6. Robots must be able to move around the room (NO). _____
7. Robots are operated using remote controls (NO). _____
8. People tell robots how to behave with a list of instructions called a program (YES). _____
9. Some robots can tell what is going on around them (YES). _____
   (Examples: sensing light, temperature, sound, or a touch.)
10. Robots are alive (NO). _____ *

*note: Understanding that robots are not alive is a tricky concept for some children. It can be especially confusing when building robotic animals that are made to resemble living things. This is a good point in the curriculum to reinforce the characteristics of living and non-living things, and looking at the differences between robotic animals and living animals.

3) Discussion: What is a robot? As a class, children discuss what they think a robot is and examples of robots they know of. Children and teachers can bring in pictures of these objects later and put them on the “Is It a Robot?” chart. The teacher shows a pre-built robotic animal a non-robotic toy animal. The class identifies that you have to push the non-robot to make it move. You can also push the robot, but (as the teacher shows) you can give it instructions and push a button to make it follow them. Why can the robot do this? It has special parts, which the teacher overviews now.

**Building and Programming a LEGO® Robot**

**Introducing the concepts and task:** Build robotic animals that are programmed to move.

1. Introduce the robot’s key parts and their functions.
2. **Communication with a robot:** Explain that we can tell a robot what to do, as long as we use a language it understands. Encourage the students to offer examples of how people communicate (speaking, writing, drawing, facial expressions, etc) and other languages they (or people they know) can speak. Discuss the idea of translating between languages, and the need to translate what we want a robot to do into the robot’s language. A *program* is another word for instructions we give the robot.

3. **Show how to use the programming interface on the computer.** Briefly describe the icons (children will learn more about programming in the next lesson). In this lesson, children will solely concentrate on programming their motors to move in order to test their robotic creations.

4. **Individual/pair work:** Students build their own robotic animals. When planning how to build and program their robot animal, students will think about the animal’s physical and behavioral attributes. Allow the students to build how they see fit, but remind them that a working robot must have a computer ‘brain’, motors, properly connected wires, and an unobstructed IR receiver. When they think they have a working robot, they bring it to a testing station where they upload the program “Begin, Forward, End” and run it. This test is to ensure that their robot follows the instruction properly and that it is sturdy. Teachers can help make sure the robots’ wires are properly oriented so that the motors turn as expected to make the robot go forward. Have the students name their robots!

**Biology Connection:** Comparing and Contrasting Different Groups of Animals

Before children choose which type of animal they would like to build, expose them to many different types of animals through books, field-trips, and activities. As a class, compare and contrast groups of animals (e.g., insects, birds, fish, mammals) and look at how animals in these groups are more similar to one another than to animals in other groups. When children choose an animal to focus on in this lesson, have them recall this knowledge and identify what group their animal belongs to. How have they designed their animal to look like a member of this group (e.g. if it is a type of bird, does it have wings?)

**Lesson 2 Vocabulary**

* **Automatic** – by itself, without help from a person
* **Computer** – a machine that gives a robot its program or instructions
* **Function** – the reason a machine or robot was built
* **Motor** – the part of a robot that makes it move
* **Robot** – a machine that can be programmed to do different things
* **Wheels** – the round parts of a vehicle that turn in circles and allow it to move
* **Wires** – the long, skinny tubes that connect all the robot’s parts

*you could also incorporate different types of animals into this lesson’s vocabulary*
To Upload from the Tangible Programming Blocks
Place the program in the camera’s field of view, and place the robot’s ‘ear’ in front of the IR tower. Click the “Upload from blocks” button. The computer takes a photo to “see” the blocks, and shows us the program on the screen.

To Upload from the Graphical Icons
Click and drag the icons together; they “snap together” when they are close enough and can make a logical sequence. Place the robot’s ‘ear’ in front of the IR tower. Click the “Upload from screen” button.

Finally, in either case, the computer sends the program to the robot. When the robot beeps, it has successfully downloaded the program. Press the green “Run” button on the RCX to make it do the program. If the robot doesn’t move as expected, brainstorm why that might be and try to fix it.

Really, when we click on an “Upload” button, the computer reads the circular codes on the icons and translates the program to robot language, which is made of 0’s and 1’s! Then the computer sends the message to the robot using infrared light. ... That’s a lot of translating between many special kinds of languages from our idea to a robot moving! (This may also be too much information for many students.)

Language Arts Connection: How-To Guide
You need to explain to someone how to build a robot animal the way you did. In your Engineering Design Journals, create a series of drawings showing all the different robotic and non-robotic parts you used. Try to use the new vocabulary words you’ve learned to label the different parts, or dictate to a teacher who can write the labels down for you.

Robotics Experts: Children who finish building their robots and master all concepts quickly get to wear a badge that says “Robotics Expert”. Robotics Experts walk around and offer help to any classmates experiencing difficulties building a functional robot.

Collaboration Web: As children progress through the lesson, they will complete their collaboration webs. They will draw lines from their picture to the pictures of any classmates who give them help. If children say they didn’t receive any help, remind them to think of their partners, class Experts, or if they got any ideas by looking at another classmate’s project.

Technology Circle: Have the students share their creations with the rest of the class (or a small group). During this time, students can share the parts and features of their robot, share what they found easy or difficult, or share what makes their robot sturdy. What do you think will happen if you make a robot that is missing one of its pieces? Try it out!

Concluding activity: See Appendix B for examples.

Free-play: Free exploration of building and programming with robotic materials. Children may choose to continue modifying their animals or explore the building and programming of other creations.
Overview: Children program their robot animal to do the Hokey Pokey.

What Is a Program?

A program is a sequence of instructions that the robot acts out in order. Each instruction has a specific meaning, and the order of the instructions affects the robot’s overall actions.

<table>
<thead>
<tr>
<th>Prior Knowledge</th>
<th>Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Students will understand that…</td>
<td>Students will be able to…</td>
</tr>
</tbody>
</table>
| • A robot is a machine that can act on its own once it receives proper instructions. | • Each icon or “block” corresponds to a specific instruction
• A program is a sequence of instructions that is followed by a robot
• The order of the blocks dictates the order in which the robot executes the instructions | • Point out or select the appropriate block corresponding to a planned robot action
• Connect a series of blocks by fitting the pegs of one block into the hole of the following block
• Upload a program to the computer and transmit it to a robot |

Materials / resources:

- Large icons for games and reference displays
- One working robot, built in previous lessons, for each child or pair
- Computers with CHERP software

Activity description

Warm-Up: Play Simon Says or another game from Appendix B to learn/review each of the programming icons and what each icon represents.

Introduce the concepts and task. Show an example robot and have the class name it. “Today we will give instructions, or programs, to our robotic animals so they will do the Hokey-Pokey.” The whole class sings and dances the Hokey Pokey to make sure everyone remembers it. Include a few verses where the children dance as different types of animals (See Biology Connection below). Conclude with a “robot verse”:
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.)

**Biology Connection: How Do Different Animals Move?**

How do different animals move? For example, in what ways does a bird move differently from a dog? Or a horse? What physical characteristics allow different animals to move in these ways (e.g. wings, hooves, tails, paws, etc.)? Dance the Hokey Pokey as different types of animals!

**Activity:** Individually or in pairs, students program their robot to do the Hokey Pokey dance. When all groups are done, everyone does the Hokey Pokey with the robot animals!

**Extension:** After you program your robot to dance the Hokey Pokey, think about how you can modify the program so that it moves more the way your specific animal would move (e.g. if your animal would be noisy, program in some extra “Beeps” or “Sings”; if your animal is bouncy, try adding in some more “Shakes”)

**Math Connection: Counting and Sequence**

How many times did you use each programming block? What order did you put your blocks in? Children will keep track of the number of forward, backward, spin, shake, beep, and sing blocks they use. Did the whole class use the same number of each block? In the same sequence?

**Lesson 3 Vocabulary:**

- **Instruction** – a direction that a robot will listen to
- **Order** – parts of a group arranged to make sense
- **Program** – a set of instructions for a robot
- **Sequence** – the order of instructions that a robot will follow exactly

**Language Arts Connection: Program Charades**

Children will pair up. One child will make up a program using the CHERP icons and act it out while the other partner guesses what the programming instructions are. Switch roles. Come up with a program together that you will “write” out (using stickers or cutouts of the CHERP instructions) to act out for the class.

**Programming Experts:** Children who finish programming their robots and master all concepts quickly get to wear a badge that says “Programming Expert”. Programming Experts walk around and offer help troubleshooting to any classmates experiencing difficulties programming.

**Collaboration Web:** As children progress through the lesson, they will complete their collaboration webs. They will draw lines from their picture to the pictures of any classmates who give them help. If children
say they didn’t receive any help, remind them to think of their partners, class Experts, or if they got any ideas by looking at another classmate’s project.

**Technology circle:** At the end of class, children will share their robotic animals with the group and talk about their process making and programming them. As a class, they will help troubleshoot problems if any children could not get their robots to function properly. Draw attention to what order to put the blocks in. Is it the Hokey Pokey if the right blocks are in a different order?

**Free-Play:** Students continue to create and upload programs to a robot. As students are ready, prompt them to plan ahead about what they want the robot to do.
Lesson 4
What Are Repeats?

Overview

Students will learn about a new instruction that makes the robot repeat other instructions infinitely or a given number of times. They use these new instructions to program their robots to move between two destinations on opposite ends of a “road” with a turn in it.

<table>
<thead>
<tr>
<th>Prior Knowledge</th>
<th>Objectives</th>
<th>Students will be able to...</th>
</tr>
</thead>
</table>
| • Arranging the same blocks in a different order will result in a different program. | • An instruction or sequence of instructions may be modified to repeat.  
• Some programming instructions, like ‘Repeat,’ can be qualified with additional information. | • Recognize a situation that requires a looped program.  
• Make a program that loops.  
• Use number parameters to modify the number of times a loop runs. |

Materials / resources:

• Large icons for games and reference displays  
• One working robot, build in previous lessons, for each child or pair  
• Computers with CHERP software  
• “Home” and student-made destination icons or models for the robots’ trips

Activity description

Warm-Up: Game or song that uses repetition. See Appendix B for examples.

Introduce the concepts: Repeats

1. Discuss what it means for something to repeat. How does this relate to similar concepts like patterns?
2. Introduce the “Repeat Forever” and “End Repeat” blocks. What does it mean to repeat something? Make a model repeating program to demonstrate the proper syntax. What will the robot do? Emphasize that the robot only repeats the instructions in between the “Repeat” and the “End Repeat” blocks. The robot does the program in order: any instruction before the “Repeat,” the repeating chunk, and then any instructions after the “End Repeat.”
3. Make a road or path of tape on the floor, and have the students choose the destination at its end (perhaps the animal’s food bowl, or its home). All together, build a program to make the robot drive along just the long portion of an “L” shaped road, but with no number parameter: [Begin, Repeat Forever, Forward, End Repeat, End].
4. Upload the program to a robot and run it. The robot drives past the end of the road. Ask the students to think about how to resolve this issue. Introduce the Number Parameters (“Numbers”)

40
and model how to add them to the repeating program so that it loops the given number of times before stopping. Now what happens to instructions placed before “Repeat” or after “End Repeat”?

5. Use the display icons to post the program (including the Number Parameter) visibly in the room.

### Lesson 4 Vocabulary:

**Loop** – something that repeats over and over again  
**Parameter** – a limit that a robot will follow  
**Pattern** – a design or sequence that repeats  
**Repeat** – to do something more than once

### Math Connection: Patterns & Counting

After showing a robot acting out a sample program that is a pattern, children will identify the repeating unit, count how many times it repeats, and (as a class) change the program so that it uses a repeat to accomplish the same outcome.

### The task: Animal Trips (individual/pair work)

Students explore a situation in which some but not all the instructions need to be repeated. The students program their robotic animals to move from one destination to another along an “L” shaped road, making the robot stop when it arrives (stopping close to the destination icon counts!). Set up several roads, perhaps of different lengths, with one leg of each road being at least 2-3 “Forwards” long. Encourage students to be creative in coming up with ideas about where their animal is traveling and why (see biology connection below).

### Biology Connection: Animal Migration

Before programming their robotic animals to go on this journey, have a lesson on animal migration.

**What is Animal Migration?** Animal migration is the relatively long-distance movement of some animals, usually on a seasonal basis.

When children program their robot animals to travel in this lesson, have them come up with an explanation of where and why their animals are traveling. Is their animal a type that migrates? Or is their animal looking for food? Or is it a pet being taken for a walk around the block? Make sure children understand the differences between animal traveling/movement and animal migration.

### Notes:

Adaptation: Break the challenge into parts: first have students program their robots to drive along one part of their road before adding the turn and the second leg of the journey. Such adjustments can make a big difference for some students as using Repeats can be complex.
**Language Arts Connection: Toothbrush Exercise**

Think about the way you brush your teeth—this is a task that requires some repeating s (like moving your toothbrush from left to right) and other motions that only happen once (like squeezing out toothpaste). Pretend YOU are a robot that needs a program to brush your teeth. Using CHERP programming instructions (and made up instructions like “spit” and “rinse”). Make up a program that uses repeats and act it out for a partner. Did you have the same program or different programs?

**Repeats Experts:** Children who finish programming their robots and master all concepts quickly get to wear a badge that says “Repeats Expert”. Repeats Experts walk around and offer help troubleshooting to any classmates experiencing difficulties programming with repeats.

**Collaboration Web:** As children progress through the lesson, they will complete their collaboration webs. They will draw lines from their picture to the pictures of any classmates who give them help. If children say they didn’t receive any help, remind them to think of their partners, class Experts, or if they got any ideas by looking at another classmate’s project.

**Technology Circle:** Students share their programs and discuss how Repeats work, especially how order is important.

**Concluding Activity:** Song or game. See Appendix B for examples.

**Free-play:**

Students need to explore the new instructions. They should build programs that use (or don’t use) them. In doing so, they will gain comfort with sequencing the blocks correctly, how the robot follows instructions before, between, or after the “Repeat” and “End Repeat” blocks, and when Repeats are helpful to use.
Overview

The students program a robot animal to perform an activity when it is light and go to sleep when it is dark.

<table>
<thead>
<tr>
<th>Prior Knowledge</th>
<th>Objectives</th>
<th>Students will be able to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Examples of human or animal sense organs and that people and animals use information provided by their senses to help make decisions.</td>
<td>• A robot can feel and see its surroundings with a sensor.</td>
<td>• Connect a light or touch sensor to the correct port on the RCX.</td>
</tr>
<tr>
<td></td>
<td>• A robot can react to collected data by changing its behavior.</td>
<td>• Write a program that includes waiting for a specific condition.</td>
</tr>
<tr>
<td></td>
<td>• Certain instructions (like “Repeat”) can be modified with sensor data.</td>
<td></td>
</tr>
</tbody>
</table>

Notes:

The LEGO® sensors detect light vs. dark or touched vs. not touched. CHERP uses this information in a binary way: “Do I see light?” or, “Is the touch sensor button pressed?” Yes or no? CHERP does not use information about degrees of brightness or pressure.

Sensors MUST be attached to Port 1 (one at a time) to work. Sensors must be triggered at the right moment. Set the sensor (press the touch sensor or create the desired light) before running the program.

The light bulb MUST be attached to Port B, but it can be attached by a wire and placed elsewhere on the robot. You need to program the light to turn off at the end of the program or else it will remain on, even when you press Run again. Alternatively, you can turn the robot on and off before rerunning the program.

The light sensors can be finicky in different lighting. TEST OUT THIS ACTIVITY BEFORE HAVING STUDENTS TRY IT. If the light sensors don’t work well in your classroom, it is possible to do this lesson using touch sensors. Lesson 6 also uses sensors, so it is reasonable to also choose just one of Lessons 5 or 6 to do.

Materials / resources:

• Large icons for games and reference displays
• One working robot, build in previous lessons, for each child or pair
• Computers with CHERP software
• A cardboard box (or other material) to create a dark tunnel for the robots to drive through
• A LEGO® light, light (or touch) sensor, and 2 short wires for each robot
Activity description

**Warm-Up:** Game or song that uses the 5 human senses. See Appendix B for examples.

**Introduce the concepts:** Sensors and Sensor Parameters:

1. Discuss examples of human / animal senses and how these senses let us gather information about what’s going on around us, so that we can make decisions based on this information.
2. Show the Light Sensor and explain how it works. (It detects light, but is not a camera. It tells the robot if it's light or dark out, but doesn’t tell it what to do.) How might this be useful? Add these to the Robot Parts poster if one is being used.
3. We need programming instructions to tell the robot what to do with the information from its sensors. Show the Repeat blocks, which are now familiar, and the new Until Light/ Until Dark blocks. Create an example program together, such as: [Begin, Repeat Until Light, Shake, End Repeat, Spin, End]
4. Run the program using an example animal, and have students discuss what the robot is doing. For example, a dog shakes to wag its tail during the day, and stops when it is dark to go to sleep.
5. Display the reference program visibly in the room.

**The task:**

The students add a light and a sensor to their robot and program their animal to perform an activity when it is light, and go to sleep (stop) when it is dark. (You can use either sensor)

1. The final program could look something like: [Begin, Repeat Until Dark (or Pushed), Shake, End Repeat, End]

**Biology Connection 1:** Human and Animal Senses

If you have small animals in the classroom, observe them while they find food, water, shelter, etc. What senses are they using? Talk about how people and animals use their senses every day.

**Math Connection:** Counting

After children add light sensors to their robots, they get together with partners. Using an “Until Light” program, one partner starts the program and stops it by activating the light sensor with a flashlight. Meanwhile, the other partner counts and records the number of times the robot does the motion (forward, spin, etc) before it stops. Partners take turns in the different roles. This activity can also be done with a touch sensor.
**Lesson 5 Vocabulary:**

- **Direction** – the way something is pointing
- **Motion** – the state when something is moving
- **Power** – the speed at which a motor moves
- **Sensor** – a machine that can tell something that is happening around it
- **Vision** – the sense used by the eyes

**Language Arts Connections: Sensor Walk**

Divide the class into two groups: Humans and Robots. Take the class for a walk around the school or neighborhood. As a class, keep a list of all the different things the humans and robots can sense and what part they used to sense it. For example, the human group may sense the sunlight with their eyes while students in the robot group would sense this with their light sensors. Children in the robot group do not need to be limited to CHERP sensors, but can think creatively about all kinds of sensors a robot might have. Upon returning to the classroom, compare and contrast the Human and Robot lists. Are there some things humans can sense but robots cannot? What about vice versa?

**Sensors Experts:** Children who finish programming their robots and master all concepts quickly get to wear a badge that says “Sensors Expert”. Sensors Experts walk around and offer help troubleshooting to any classmates experiencing difficulties attaching their sensors or programming with sensors.

**Collaboration Web:** As children progress through the lesson, they will complete their collaboration webs. They will draw lines from their picture to the pictures of any classmates who give them help. If children say they didn’t receive any help, remind them to think of their partners, class Experts, or if they got any ideas by looking at another classmate’s project.

**Technology Circle:** Understanding how the sensors work and how programming them can be challenging. Have students discuss their understandings of the sensors and why different programs do or do not accomplish the goal.

**Concluding Activity:** Song or game. See Appendix B for examples.

**Free-play** Have students explore adding sensors to robots and making programs with Repeats and Sensor Parameter instructions. Sensors use complex concepts and often work in unexpected ways. Offer support in observing the robot’s behavior so students may fully understand these concepts.
Lesson 6
What are “Ifs?”

Overview:

Students program a robot animal to take different actions based on the state of a sensor.

<table>
<thead>
<tr>
<th>Prior Knowledge</th>
<th>Objectives</th>
<th>Students will be able to...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Students will understand that...</td>
<td></td>
</tr>
<tr>
<td>• Some instructions can be qualified with additional information.</td>
<td>• A robot can ‘choose’ between two sequences of instructions depending on the state of a sensor.</td>
<td>• Connect a light or touch sensor to the correct port on the RCX.</td>
</tr>
<tr>
<td>• A robot can feel and see its surroundings with a sensor.</td>
<td></td>
<td>• Identify a situation that needs a branched program.</td>
</tr>
<tr>
<td>• A robot can react to information it collects by changing its behavior.</td>
<td></td>
<td>• Make a program that uses a branch.</td>
</tr>
</tbody>
</table>

Materials / resources:
• Large icons for games and reference displays
• One working robot, built in Lesson 2, for each child or pair
• Light or touch sensor, a short and long wire for each robot
• Computers with CHERP software, webcams, IR towers, programming blocks
• “Home” and destination icons or models placed on the floor, tape roads

Activity description

Warm-Up: Game or song. See Appendix B for examples.

Introduce the concept: “If”

1. In the programs so far, the robot has only one choice of what instructions to do next. Today we will learn an instruction that give the robot two choices, and the robot uses a sensor to know which set of instructions to follow each time the program is run. Solicit examples of times we rely on sensors to help us make decisions. (If I feel something prickly, I’ll move away from it. Or, if I see it’s rainy out, I’ll bring an umbrella; if not, I’ll leave the umbrella at home.)
2. Play “Simon Says” to help the students gain familiarity with the thought process behind branches. For example, “Simon says, ‘If the lights are on, jump twice, (if not, stand on one foot).’”
3. Introduce (or reinforce) the concept of nocturnal animals that are awake at night. Make a list of examples of nocturnal animals and their typical behaviors.
4. Introduce If, End-If blocks and light/dark and pushed / released parameters. Make and act out a model program together based on a nocturnal animal. (If dark, go forward to search for food, shake to eat, etc...). Upload it to a robot to see how it works. It is best to start out with a program that uses only “If” and to save the “If Not” segment for an extension if students are ready for it.
Students will have a much better understanding of how “If” gives the robot choices once they have run the program themselves in both sensor conditions.

5. Once the students run the program, use the reference icons to post the program in the room.

**The task: The Robot Animal Chooses a Program**

1. Students will program their robot animal to do perform one behavior if dark and another if light.
2. Extension: Students program their robot to follow one branch of a T-shaped map or another depending on light sensor input. The robot starts at the base of the T, and the sun is shining (or not) at it. If it’s still ‘light out,’ the robot has time to go play at the park (to the right). If not, the robot should go right home (to the left). (This can be adapted to use a touch sensor.)

**Biology Connection: Diurnal vs. Nocturnal Animals**

*Children discuss and share ideas about the sleeping cycles of different types of animals. Children learn to differentiate diurnal from nocturnal animals and use this knowledge when programming their robotic animals to do one thing if it is light out and another if it is dark out.*

**DIURNAL ANIMALS** are awake during the day and asleep at night

**NOCTURNAL ANIMALS** are asleep during the day and awake at night.

**Math Connection: Red Light, Green Light**

*In small groups, have students take turns being the “Traffic cop.” The Traffic cop gives out orders to the group such as “If green, go jump 3 times. If red, sit down.” The Traffic cop then holds up either a red or a green piece of paper, and the other students in the group must complete the instructions accordingly. Try the game as a class first, and once the children feel comfortable with it, allow them to break into smaller groups and try being the Traffic cop themselves.*

**Literacy Connection: If Worksheet**

*Students will complete a worksheet about their daily and weekly schedules. They will fill in the second half of an “If, then” statement such as “If it is Saturday, _________” in any way they want and draw a picture of the activity they’re describing.*

**Ifs Experts:** Children who finish programming their robots and master all concepts quickly get to wear a badge that says “Ifs Expert”. Ifs Experts walk around and offer help troubleshooting to any classmates experiencing difficulties attaching their sensors or programming with ifs.

**Technology Circle:** Students share the program they made, what it does, and anything they found easy, hard, or surprising during the activity. Children sometimes think that Ifs make the robot do one program
or the other whenever the sensor is in that state rather than as a one-time decision-maker for which set of instructions the robot will follow. This is important to identify and clarify with demonstrations.

*Concluding activity*: Song or game. See Appendix B for other suggestions.

*Free-play* Let students explore building programs with the If blocks. This exploration gives them a chance to learn how to use the block in a program, think of situations that require it, and further understand how to use sensors.
Lesson 7
The Animal Project

Overview

This project should be tailored to fit with a curriculum unit on animals so that it meets the goals of the teachers and the interests of the students and teachers. Students work alone or in pairs to build and program a robot to demonstrate their understandings and ideas related to the robotics and programming curriculum as well as an animal of their choice. During the course of the final project, students put to use all the concepts learned during the previous lessons but transfer them to a new context. When possible, teachers should encourage the use of crafts and recycled materials.

**Note:** The work for the final project should be broken up into several sessions. It is up to the teacher when to complete each part of the project. Not all of the activities need to be completed during Robotics time. It is left up to the teacher’s discretion whether the students will build a new sturdy robot from scratch or will use the robots they build in earlier lessons.

**Language Arts Connection:**

- **Invitations:** Write out and mail invitations to your family inviting them to come to your final project presentation. Add illustrations and information describing your project.

- **How-To Book:** Create a comprehensive How-To Book describing how to build and program the robot you made

**Math Connection:**

**How Many?** As a class, keep a chart that graphs how many of all the different types of robotic and non robotic parts you used. Make a report to display and share on the presentation day.

**Individual work:** Children will work individually to plan, design, build, and program a final project from scratch. Children will be encouraged to use advanced topics such as sensors and repeats when programming their robots.

1. Each child or pair will choose one animal and research that animal. In their Engineering Design Journals, they should write down five facts about that animal. This activity should be done with the help of an adult or older child. For example, one classroom did this activity with their 4th Grade reading buddies.
2. Using their skills from Lessons 1 and 2, each child or pair will build a sturdy robot, keeping in mind that this robot will eventually become the animal they chose.
3. Children will use the facts they gathered and their programming skills in order to plan and create a program for their animal. They should try to incorporate behaviors and movements specific to that
animal. Students will need to be creative in order come up with ways to make the CHERP robot instructions representative of the real behaviors of their animals. Once they have a program they are happy with, they should record it in their Engineering Design Journals, and an adult should check it.

4. The children will decorate their robots to look like their animals using a combination of LEGO® and arts and crafts materials. It is important that the children do not obstruct the IR receiver or cover the buttons they will need to use their robots.

5. The children will create a “habitat” for their animal by drawing on a large piece of paper and/or building LEGO® structures. This will be used as a background for their final presentations.

6. Using their skills from Lessons 1 and 2, each child will build a sturdy robot and test it.

7. Children will design a name card to attach to their robot. Children will write their name on an index

<table>
<thead>
<tr>
<th>More Animal Project Ideas:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Use the light sensor to create a program that shows how the animal behaves during the day and at night.</td>
</tr>
<tr>
<td>- Find ways to make the robot move like your animal (e.g. turn right, turn left could be like a slithering snake).</td>
</tr>
<tr>
<td>- Include sounds as well as movements (e.g. an owl could Beep to hoot and Spin to spin its head).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Presentations: Students share:</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. the robot animal they made,</td>
</tr>
<tr>
<td>b. why they chose the features they did for their robot,</td>
</tr>
<tr>
<td>c. the goal of their program and why they wanted it to do that / what it represents,</td>
</tr>
<tr>
<td>d. the final program they built, and</td>
</tr>
<tr>
<td>e. anything that was hard, easy, surprising, interesting, etc about the process.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Materials / resources:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Large icons for games and reference displays</td>
</tr>
<tr>
<td>- Robotic parts for each child to make a robot, plus extras</td>
</tr>
<tr>
<td>- Crafts and recycled materials for robots and for building an environment for them to run in</td>
</tr>
<tr>
<td>- Computers with CHERP software,</td>
</tr>
<tr>
<td>- Design journals, small icons for cutting and taping/gluing in the design journals</td>
</tr>
<tr>
<td>- Large posters for a class timeline</td>
</tr>
<tr>
<td>- Photos/drawings of events throughout the year</td>
</tr>
</tbody>
</table>
Appendix A
Robotics across Themes
**Robotics across Themes**

This robotics and programming curriculum can be used within the context of study on a wide variety of topics. The challenges presented in this curriculum relate to the theme of animals, but the basic ideas can be reconfigured to make sense with many other topics commonly studied in early childhood settings. Below are some suggestions for such contexts. These ideas may also help children focus in choosing what kind of robot to make if the curriculum uses an open-ended theme such as “communities.”

- Person who lives in the community
- Person who works in the community
- Responsive or interactive building or structure
- Vehicles and road-related structures (traffic lights, drawbridges, etc)
- Nature (plants, animals, landscape, weather)
- Safety (alarms, crossing signals)
- Places and structures for entertainment, fun, or commerce
- Basic needs (food, housing)

The activities in this curriculum can easily be adapted to fit other themes. Use your imagination to find a story context for each powerful idea. For instance, if this curriculum were to accompany a unit on transportation, the curriculum might look like this:

**Lesson 1:** (What is the Engineering Design Process?) Build a sturdy vehicle that can move like its real-life counterpart.

**Lesson 2:** (What is a Robot?) Build a robotic car.

**Lesson 3:** (What is Programming?) “The Hokey Pokey” (or another of your choosing) is a fun and concrete way to start the unit regardless of the theme.

**Lesson 4:** (What are Repeats?) Program your car to visit different locations on a map using repeat parameters to simplify your program.

**Lesson 5:** (What are Sensors?) Add a touch sensor and a light to your car and program it to turn on when it reaches a dark part of the road.
Appendix B
Songs and Games
Many common songs and games can be used to support children’s understandings of robots and programming concepts. For instance, Simon Says is a way to internalize instructions and understand them in a more complete way, both kinesthetically and verbally. Here are some other suggestions for songs and games to reinforce various concepts from the curriculum. Teachers may think of many more!

- Simon Says, traditional style: emphasizes ways our own bodies move, but without having kids sit out for mistakes,
- Simon Says w/ icons cards: helps students learn new programming icons’ symbols, spoken name, and kinesthetic action. Variation: Kids pick icons/strings for peers to act out, Head, Shoulders, Knees, and Toes: emphasizes peoples’ body parts vs. robot parts,
- Act out blocks and programs or ‘program’ a friend to move along a line on the floor,
- ‘Memory’ card game or other matching game with icons: spurs use of instruction icons’ names. When a child finds a match, they name and act out the icon.
- ‘The Wheels on the bus.’ Variation: Sing with programming instructions,
- Programming Charades: Mentor shows a child a program made from block icons. The child acts out the program. The other children identify what icons made up the program.
- Walk-Through Programs. Make large programming icons that can be placed on the floor. Children literally walk through the program step by step and carry out the actions to internalize how the robot processes its programs. This can be especially helpful when working with “Repeats” and “Ifs.”
Appendix C
The Engineering Design Process
The Engineering Design Process

When working with young children and robotics, there are some interesting challenges around helping children structure their problem-solving processes. Marina Bers, in her Blocks to Robots’ book (2008), talks about the role of the engineering design process.

“On the one hand, we want to help them [the children] follow their ideas, but we do not want them to become frustrated to the point they quit the work. On the other hand, we do not want their success to be scripted, too easy, or without failure. One of the approaches for how to handle this is by helping them understand and follow the design process. This is similar to what engineers or software developers do in their own work. They identify a problem. They do research to understand better the problem and to address it. They brainstorm different potential solutions and evaluate the pros and cons. They choose the best possible solution and plan in advance how to implement it. They create a prototype and they implement it. They test it and redesign it based on feedback. This happens many, many times. And finally, they share their solutions with others. This cycle is repeated multiple times.”

The following diagram, Figure 1, shows one of many possible simplified versions of the engineering design process. One suggestion for using this graphic is to give each child or pair a small copy of it along with a token, similar to a playing piece in a board game. Children can move their token around the diagram to reinforce the steps of the process. Figures 2-7 show individual steps of the engineering design process to facilitate making a large poster of the whole process for the class to see from anywhere around the room.
Figure 1: Simplified steps involved in the engineering design process.
**Figure 2:** Engineering Design Process step 1: Ask a question about a problem you want to solve or a goal you want to accomplish.
Figure 3: Engineering Design Process step 2: Imagine as many different ways to accomplish your goal or answer your question as you can.
Figure 4: Engineering Design Process step 3: Choose one solution and plan out how to do it in detail.
Figure 5: Engineering Design Process step 4: Create a prototype or working version of your plan.
Figure 6: Engineering Design Process step 5: Test your creation to see how well it accomplishes the goals you have for it. Try different ways to improve it and test whether the improvements work better.
Figure 7: Engineering Design Process step 6: Share what you have done and get feedback.
Appendix D
Design Journals
**Design Journals**

Providing children with a design journal and with many opportunities to talk about their ideas throughout the process can be helpful. However, before working with design journals it is useful to be aware of different approaches to the design and problem-solving processes. Following is an excerpt from Marina Bers’ book *Blocks to Robots*.

“As children work on their projects, many iterations and revisions will be done. Design journals make transparent to the children themselves, as well as teachers and parents, their own thinking and the project evolution. [...] Some children might choose to avoid using design journals or follow a systematic design process. They do not like to plan in advance. They might belong to a group of learners that Papert and Turkle have characterized as tinkerers and bricoleurs (Turkle & Papert, 1992). They engage in dialogues and negotiations with the technology, their ideas happen as they design, build and program. As Papert and Turkle write, “The bricoleur resembles the painter who stands back between brushstrokes, looks at the canvas, and only after this contemplation, decides what to do next” (Turkle & Papert, 1992).

“Constructionist learning environments allow for different epistemological styles, or ways of knowing, to flourish. Some children want and need constraints and top-down planning because they know what they want to make. Others enjoy working bottom-up and messing around with the materials to come up with ideas. Some methods of teaching robotics and programming, directly derived from engineering and computer sciences, provide structured paths for children to navigate the process from idea to product. For example, the formal steps of the engineering design process presented earlier are laid out in a design journal consisting of teacher made worksheets. This approach might or might not work, depending on the child, the way the learning environment is set up and the educational goals. In this book I advocate both pathways: design journals with a directive focus, in the forms of questions and design journals with lots of white pages, for those children that might want to invent their own strategies. Tinkers and planners complement each other and can also learn from each other. Constructionist environments should be inviting and supportive to little engineers who thrive working with constraints and making advanced plans, and little tinkerers who create in dialogue with the materials.”
Taking these ideas into consideration, a sample design journal (with math and language arts activities tailored to PreK-Kindergarten children) begins on the next page. This journal is a template that can be printed and used directly by teachers or modified to fit their specific goals. Additionally, activities are provided for the advanced lessons (sensors and gears) and should be taken out of design journals for children not pursuing these lessons.
Name:________________________

MY ENGINEERING DESIGN JOURNAL

The Engineering Design Process

Ask

Imagine

Plan

Create

Test & Improve

Share

This is what we made!
Lesson 1: Trip to the Park

I picked the ____________________.

It looks like this:

It has these shapes:
Lesson 1: Building Something from the Park

I’m building a ________________________________.

Here’s a picture of what it will look like:

To build it, I will use these shapes:
Lesson 1: Postcard

Choose a friend or family and write them a postcard describing what you saw and built today. Cut out this postcard and draw a picture of what you built on the back. Then write the name and address of the person you want to send it to, put on a stamp, and send it!

Try to use some of these words:

Artifact    Design    Material    Square
Circle      Edge      Rectangle   Vehicle
Cycle       Engineer  

Dear __________________,

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

________________________

70
Lesson 2: Planning My Robotic Animal

Here is a picture of what my robot will look like:

![Robot Image]

These are the shapes I will use to build it:
Lesson 2: My Robotic Animal (continued)

I used these pieces to build my robotic animal:

<table>
<thead>
<tr>
<th>Piece:</th>
<th>How many:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total number of pieces:
Lesson 2: My Robotic Animal (continued)

I used these colors to build my vehicle:

<table>
<thead>
<tr>
<th>Color:</th>
<th>How many:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I used these shapes to build my robot:

<table>
<thead>
<tr>
<th>Shape:</th>
<th>How many:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

73
Lesson 2: How to Build a Robotic Animal

The robot I built is a ____________________.

Here's how I did it:

<table>
<thead>
<tr>
<th>Words to Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic</td>
</tr>
<tr>
<td>Axle</td>
</tr>
<tr>
<td>Computer</td>
</tr>
</tbody>
</table>
Lesson 2: How to Build a Robotic Animal (continued)

This is what it looks like:
Lesson 3: The Hokey-Pokey

Here's the program I made to make my robot do the Hokey-Pokey:
Lesson 4: Brushing My Teeth

Here’s my program for brushing my teeth:
Lesson 4: My Animal’s Trip

Here’s a map of the trip my robot will take:

[Blank map]

Here’s the program I made:

[Blank program]
My animal is a ____________________.

It uses these senses:
______________________________________________________
______________________________________________________

I added these sensors to my robot:
______________________________________________________
______________________________________________________

Here is the program I made:

```
Lesson 6: Ifs

Finish these sentences and draw pictures describing what you do:

If it is Monday, _________________________________.

If it is Saturday, _________________________________.

If it is my birthday, _______________________________.
Lesson 7: The Animal Final Project (page 1)

The animal I picked is a __________________________.

Here’s a picture:

[Blank]

Here are some facts I learned about it:

Fact #1:
________________________________________________________________________
________________________________________________________________________

Fact #2:
________________________________________________________________________
________________________________________________________________________

Fact #3:
________________________________________________________________________
________________________________________________________________________

Fact #4:
________________________________________________________________________
________________________________________________________________________

Fact #5:
________________________________________________________________________
Lesson 7: The Animal Project (page 2)

My robotic animal will look like this:

![Diagram of animal]

I will use these materials to decorate it:
Lesson 7: The Animal Project (page 3)

Here’s what my robotic animal is going to do:

__________________________________________

__________________________________________

__________________________________________

Here’s the program:


Appendix E
Engineer's License
**Figure 8:** Sample Engineer’s License (ready to print and use)

<table>
<thead>
<tr>
<th>Level 1 – Sturdy Builder</th>
<th>Level 2 – Robot Builder</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Level 3 – Programmer I</td>
<td>Level 4 – Programmer II</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Level 5 – Programmer III</td>
<td>Level 6 – Expert</td>
</tr>
<tr>
<td>Level 1: Builder</td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>--</td>
</tr>
<tr>
<td><img src="image.png" alt="Icon" /></td>
<td>Robot stays intact.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Level 2: Robot Builder</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image.png" alt="Icon" /></td>
<td>Robot has all attached parts. Robot is able to move.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Level 3: Programmer I</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image.png" alt="Icon" /></td>
<td>Child picks right programming icons and puts icons in the right order</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Level 4: Programmer II</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image.png" alt="Icon" /></td>
<td>Child knows when and how to use repeats.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Level 5: Programmer III</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image.png" alt="Icon" /></td>
<td>Child understands sensors and ifs and how to program them.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Level 6: Expert</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image.png" alt="Icon" /></td>
<td>See criteria for assessing final projects and overall levels of understandings.</td>
</tr>
</tbody>
</table>

Key to the icons on the sample Engineer’s License
Icons to print as stickers or cut-outs for Engineer's Licenses:
Appendix F

Positive Technological Development
Figure 9: Personal Development Trajectory

Figure 10: Sample Collaboration Web
Appendix G

Working with CHERP and the LEGO® RCX
Testing a Robot’s Motors:

It is necessary to test a newly built robot to make sure that its motors turn as expected when programmed with a Forward instruction (or another instruction with an easily verifiable outcome, e.g. NOT Shake or Spin). The wires can attach to the RCX or motors parts in four directions. Two directions will make the motors turn clockwise; the other two will make the motors turn counterclockwise. Turning a wire connection 180° will always reverse the direction the motor spins; turning it 90° does not guarantee a direction change. Once you have this orientation, you can place the motors to make any side of the RCX the “front” of the robot.

Setting up the Tangible Programming Materials

The spacing between the computer’s webcam and the tangible programming blocks is important for the computer vision to work properly. The webcam must have a direct line of sight to the blocks, the blocks should be at least 18 inches from the webcam, and the whole program must fit in the camera’s field of view. To test your set-up, upload tangible programs and check the photo that appears on the screen and the graphical version of the program that the computer saw in that picture. Some helpful hints: Mark where to place blocks relative to the webcam. This could mean placing labeled notecards, paper strips, or tape on the surfaces where you know the set-up works. Or, have 18-inch strings or paper strips available for students to measure (and mark) the distance between their webcam and blocks. Best yet, experiment with how far left and right the end of a program can go without leaving the webcam’s field of view and mark those edges on paper strips taped to the work surface. The more of this spacing that can be pre-marked, the quicker students can get to work during the activities if the equipment is not already set up.

It is also important the all the blocks be in one straight line, which is usually only tricky with the magnet parameters, which can be twisted, or children might not place them in view of the webcam, and the roped Repeat and If blocks, whose cords can block the webcam's view or misalign the blocks (if the cord is underneath them).

Using the IR Tower

The IR receiver port on the RCX (the smooth black rectangle on one end of it) must be aligned with the IR tower's transmitter. It is best to place the RCX as close as possible to the tower so the RCX does not accidentally pick up a program sent by a different tower. However, the RCX's IR port should be lined up near the green light on the tower that turns on while it transmits. This means you might have to prop up or hold the RCX to be lined up properly.
Appendix H

Building Instructions:

Starter ideas for mobile robot design
Starter Ideas for Mobile Robot Designs

There are many ways to put together a mobile LEGO® robot, but starting out can be confusing for children and adults alike. Here are some ideas, intended to inspire the exploration of different designs.

- Watch out that the wires don’t rub the tires and that the wheel or tire does not rub on other parts of the robot. This will slow the motor down or prevent the wheel from turning properly.
- With some designs, you can wrap the wire back between the motors toward the back of the RCX and up onto the ports (see example (a) below).
- Use a “slider” instead of a wheel on the front “leg(s)” of the robot. This is simpler and it allows the robot to turn smoothly. A tire in front will cause a lot of friction while the robot turns.
- Try wheels of different sizes. Try using other round parts, like LEGO® gears as wheels.
- Make sure all the robot’s parts and other LEGO® and crafts or recycled pieces are connected STURDILY. It can be frustrating and time-consuming to rebuild a robot over and over!
- Keep the IR port (“ear”) unobstructed so the robot can receive programs.

![Figure 1](image-url): Possible mobile robot designs
Appendix I

List of Materials
List of Materials

Robotics materials

- 1 set of robot parts for each child or pair, plus extras of each part: RCX “computer brain” brick, 2 motors, 3+ wires (2+ short and 1+ long), a variety of different-sized wheels, LEGO® light bulb piece; touch sensor and light sensor.
- LEGO® “slider” pieces, assortment of extra LEGO®s® and recycled materials for decoration;
- Batteries (each RCX runs on 6 AA batteries).

Programming materials:

- Computers installed with CHERP software.
- 1 set of tangible programming blocks for at least every two students, regardless of whether they are working together or separately.

Art Materials

- Various art materials including paper, scissors, markers, tape, and recyclable materials

Teaching materials:

- Posters showing robot parts (See following page)
- Poster showing the Engineering Design Process
- Chart and images for “Is It a Robot?”;
- Large icons for display / reference programs. They could be magnetic or felt-backed for magnetic whiteboards or felt-boards. Use whatever display surfaces are readily available;
- Design journals for final project and icons on paper for students to cut and tape / glue into their design journals;
- Engineer Licenses
- Expert Badges
- Assessment forms for each student.
Robot Parts

Ear
Brain (Inside)
Batteries (Inside)
Body

Motors
Wires

Sensors
Light
Suggested “Robot or Not” Pictures (Lesson 2)
Appendix J:

Assessments
Assessments for Lesson 1: What is the Engineering Design Process?  Student's name:

Level 1: Engineering Design Process through Sturdy Building

When a child attempts this license level, also assess them according to the following scale. Use NA if not applicable.

If assessing a child based on a partially complete project, note this in the “Notes” section. In this case, assess the child based on their understanding of the core concepts below and how effectively they implemented them on the part of the project they did complete.

<table>
<thead>
<tr>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
<th>Did not attempt/Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete Achievement of goal/task/understanding</td>
<td>Mostly Complete Achievement of goal/task/understanding</td>
<td>Partially Complete Achievement of goal/task/understanding</td>
<td>Very Incomplete Achievement of goal/task/understanding</td>
<td>Did Not Complete goal/task/understanding</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Skill</th>
<th>Achievement Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Structure is built to add and remove a toy person</td>
<td>5 4 3 2 1 0 NA</td>
</tr>
<tr>
<td>2. Structure remains intact while being handled and functions as it is designed to function.</td>
<td>5 4 3 2 1 0 NA</td>
</tr>
</tbody>
</table>

Notes:

Overall Debugging:

| 1. A. Recognizes that something is not working. | 5 4 3 2 1 0 NA |
| 2. B. Keeps original goal or changes to an acceptable alternative. | 5 4 3 2 1 0 NA |
| 3. C. Has a hypothesis of the cause of the problem. | 5 4 3 2 1 0 NA |
| 4. D. Attempts to solve the problem. | 5 4 3 2 1 0 NA |

Notes:
Assessment for Lesson 2: What is a Robot?  

Student’s name

Level 2: Robot Builder

When a child attempts this license level, also assess them according to the following scale. Use NA if not applicable.

If assessing a child based on a partially complete project, note this in the “Notes” section. In this case, assess the child based on their understanding of the core concepts below and how effectively they implemented them on the part of the project they did complete.

<table>
<thead>
<tr>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete Achievement of goal/task/Understanding</td>
<td>Mostly Complete Achievement of goal/task/understanding</td>
<td>Partially Complete Achievement of goal/task/understanding</td>
<td>Very Incomplete Achievement of goal/task/understanding</td>
<td>Did Not Complete goal/task/understanding</td>
<td>Did not attempt/Other</td>
</tr>
</tbody>
</table>

1. Knows their robot needs specific parts for specific actions and uses those parts.

2. Attaches all necessary robot parts so that they work correctly (i.e. motor wire properly connected to hub, axle connected to motor, hub connected to computer).

3. Knows how to program the robot to move motor

Notes:

Overall Debugging:

1. A. Recognizes that something is not working.

2. B. Keeps original goal or changes to an acceptable alternative.

3. C. Has a hypothesis of the cause of the problem.

4. D. Attempts to solve the problem.

Notes:
**Assessment for Lesson 3: What is a Program?**

Level 3: Programmer I

When a child attempts this license level, also assess them according to the following scale. Use NA if not applicable.

If assessing a child based on a partially complete project, note this in the “Notes” section. In this case, assess the child based on their understanding of the core concepts below and how effectively they implemented them on the part of the project they did complete.

<table>
<thead>
<tr>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
<th>Did not attempt/Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete Achievement of goal/task/understanding</td>
<td>Mostly Complete Achievement of goal/task/understanding</td>
<td>Partially Complete Achievement of goal/task/understanding</td>
<td>Very Incomplete Achievement of goal/task/understanding</td>
<td>Did Not Complete goal/task/understanding</td>
<td>Did not attempt/Other</td>
<td></td>
</tr>
</tbody>
</table>

1. Selects the right instructions.  
2. Arranges instructions in the correct order.

**Notes:**

**Overall Debugging:**

1. A. Recognizes incorrect instructions or order by reading the program or watching the robot run the program.  
2. B. Keeps original goal.  
3. C. Has a hypothesis of the cause of the problem.  
4. D. Attempts to solve the problem.

**Notes:**
Assessment for Lesson 4: What are Repeats?

Level 4: Programmer II

When a child attempts this license level, also assess them according to the following scale. Use NA if not applicable.

If assessing a child based on a partially complete project, note this in the “Notes” section. In this case, assess the child based on their understanding of the core concepts below and how effectively they implemented them on the part of the project they did complete.

<table>
<thead>
<tr>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete Achievement of goal/task/understanding</td>
<td>Mostly Complete Achievement of goal/task/understanding</td>
<td>Partially Complete Achievement of goal/task/understanding</td>
<td>Very Incomplete Achievement of goal/task/understanding</td>
<td>Did Not Complete goal/task/understanding</td>
<td>Did not attempt/Other</td>
</tr>
</tbody>
</table>

1. Knows when and how to use Repeats.  
   5 4 3 2 1 0 NA

2. Knows when and how to use number parameters.  
   5 4 3 2 1 0 NA

3. Selects the right instructions.  
   5 4 3 2 1 0 NA

4. Arranges the instructions in the correct order.  
   5 4 3 2 1 0 NA

Notes:

Overall Debugging:

1. A. Recognizes incorrect instructions or order by reading the program or watching the robot run the program.  
   5 4 3 2 1 0 NA

2. B. Keeps original goal.  
   5 4 3 2 1 0 NA

3. C. Has a hypothesis of the cause of the problem.  
   5 4 3 2 1 0 NA

4. D. Attempts to solve the problem.  
   5 4 3 2 1 0 NA

Notes:
## Assessment for Lesson 5: What are Sensors?

**Level 4: Programmer II-B**

When a child attempts this license level, also assess them according to the following scale. Use NA if not applicable.

If assessing a child based on a partially complete project, note this in the “Notes” section. In this case, assess the child based on their understanding of the core concepts below and how effectively they implemented them on the part of the project they did complete.

<table>
<thead>
<tr>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
<th>NA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete Achievement of goal/task/understanding</td>
<td>Mostly Complete Achievement of goal/task/understanding</td>
<td>Partially Complete Achievement of goal/task/understanding</td>
<td>Very Incomplete Achievement of goal/task/understanding</td>
<td>Did Not Complete goal/task/understanding</td>
<td>Did not attempt/Other</td>
<td></td>
</tr>
</tbody>
</table>

1. Knows how to use sensors and what they are for. | 5 4 3 2 1 0 NA |
2. Knows when and how to use sensor parameters. | 5 4 3 2 1 0 NA |
3. Selects the right instructions. | 5 4 3 2 1 0 NA |
4. Arranges instructions in the correct order. | 5 4 3 2 1 0 NA |

### Notes:

**Overall Debugging:**

<table>
<thead>
<tr>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. A. Recognizes incorrect instructions or order by reading the program or watching the robot run the program.</td>
<td>5 4 3 2 1 0 NA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2. B. Keeps original goal. | 5 4 3 2 1 0 NA |
3. C. Has a hypothesis of the cause of the problem. | 5 4 3 2 1 0 NA |
4. D. Attempts to solve the problem. | 5 4 3 2 1 0 NA |

Notes:
Assessment for Lesson 6: What are Ifs?  

Level 5: Programmer III

When a child attempts this license level, also assess them according to the following scale. Use NA if not applicable.

If assessing a child based on a partially complete project, note this in the “Notes” section. In this case, assess the child based on their understanding of the core concepts below and how effectively they implemented them on the part of the project they did complete.

<table>
<thead>
<tr>
<th>Complete Achievement of goal/task/understanding</th>
<th>Mostly Complete Achievement of goal/task/understanding</th>
<th>Partially Complete Achievement of goal/task/understanding</th>
<th>Very Incomplete Achievement of goal/task/understanding</th>
<th>Did Not Complete goal/task/understanding</th>
<th>Did not attempt/Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

1. Knows how and when to use Ifs.
2. Knows how and when to use If Nots (if applicable).
3. Selects the right instructions.
4. Arranges instructions in the correct order.

Notes:

Overall Debugging:

1. A. Recognizes incorrect instructions or order by reading the program or watching the robot run the program.
2. B. Keeps original goal.
3. C. Has a hypothesis of the cause of the problem.
4. D. Attempts to solve the problem.

Notes:
**Assessment for Lesson 7: Final Project**

**Student’s name:** ____________________

**Level 6: Expert**

**Part 1: Assess each child along these scales when they have finished their final project.**

<table>
<thead>
<tr>
<th></th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete Achievement of goal/task/understanding</td>
<td>Mostly Complete Achievement of goal/task/understanding</td>
<td>Partially Complete Achievement of goal/task/understanding</td>
<td>Very Incomplete Achievement of goal/task/understanding</td>
<td>Did Not Complete goal/task/understanding</td>
<td>Did not attempt/Other</td>
<td></td>
</tr>
</tbody>
</table>

1. Has a goal for the project.  
2. Selects the right instructions to accomplish the goal.  
3. Arranges instructions in the correct order to accomplish the goal.  
4. Knows how and when to use Repeats.  
5. Knows how and when to use number parameters.  
6. Knows how to use sensors and what they are for.  
7. Knows how and when to use sensor parameters with Repeats.

**Notes:**

**Overall Debugging:**

<table>
<thead>
<tr>
<th></th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. Recognizes incorrect instructions or order by reading the program or watching the robot run the program.</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0 NA</td>
</tr>
<tr>
<td>B. Keeps original goal.</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0 NA</td>
</tr>
<tr>
<td>C. Has a hypothesis of the cause of the problem.</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0 NA</td>
</tr>
<tr>
<td>D. Attempts to solve the problem.</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0 NA</td>
</tr>
</tbody>
</table>

**Notes:**

108
Part 2: Ask each child these questions to supplement the students’ journals and the teachers’ observations of and conversations with students during work and sharing times.

- What does your program tell your robot to do? How did you choose those instructions?
- What parts does your robot have (robotic and/or aesthetic)? Why did you choose them?

Mark the students’ level of understanding of how to program a robot along the following criteria.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Units:</td>
<td>Understands the function of individual robot parts and individual programming instructions, but not how to choose and assemble them to make a functional robot or program that accomplishes a given goal.</td>
</tr>
<tr>
<td>Connections:</td>
<td>Chooses appropriate parts for the robot and instructions for the program. Puts parts together correctly and instructions in the right order. Understands that putting the parts together in certain ways creates an overall outcome. Does not see the connection between the whole program and then accomplishment of the chosen goal.</td>
</tr>
<tr>
<td>Context:</td>
<td>Understands the function of each element and that the order they are put in results in a specific overall outcome. Is able to purposefully put the right instructions in the right order for the program to achieve the given goal.</td>
</tr>
</tbody>
</table>
References


