Computational Thinking with Lego Robotics

Teacher's Guide

Developed By:

Peter Rich
Geoff Wright
Tyler Beckstrom
Jessica South

Overview of Curriculum

Purpose

We have developed this curriculum to purposefully emphasize ***computational thinking*** among elementary-aged students. Computational thinking is a way of logically solving problems, often employed by engineers, programmers, mathematics, and scientists from many different disciplines. We take a very deliberate approach to emphasizing the concepts of *variable* and *function*. We refer to the NXT programming blocks as functions to intentionally emphasize *input — output —process.* Furthermore, each lesson is arranged to engage students in 3 of the fundamental process of computational thinking. Namely:

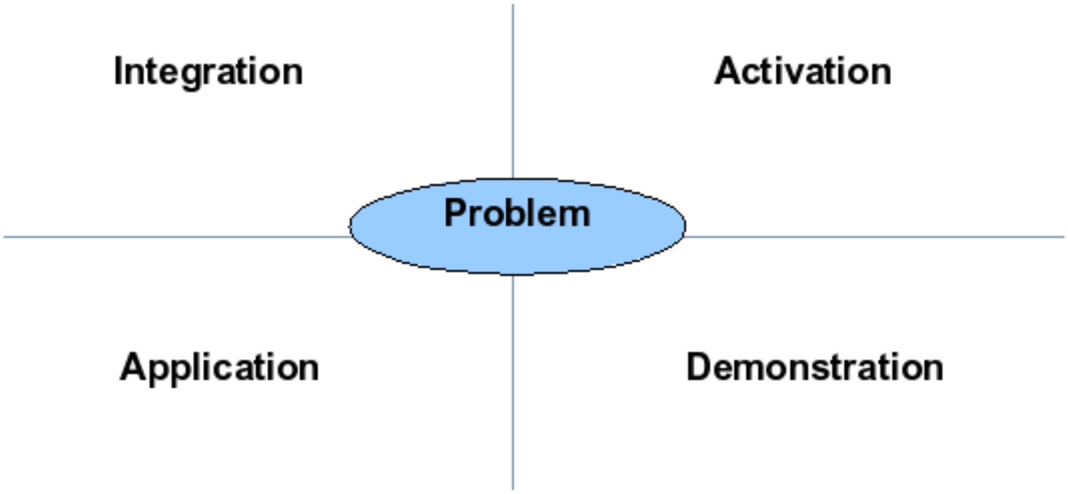
> ***Abstraction***: includes the ability to *break down* a problem into manageable parts, then to *plan* how to resolve the problem, resulting in a set of steps or an *algorithm* to follow.
>
> ***Automation***: involves *programming* the robot to carry about the plan determined in the abstraction process.
>
> ***Analysis***: involves gathering, synthesizing and interpreting data from automated processes. *Debugging* is an important analysis process that students will need to practice often.

These 8 lessons focus mostly on the programming side of the robots. If time permits, we recommend following up this unit with a unit on the physical mechanics of LEGO Robotics. Lessons in this curriculum build upon each other, gradually introducing core programming concepts. Mastering these will allow students to accomplish a majority of LEGO programming challenges. The concepts covered in these lessons are:

- Variables
- Loops
- Switches (if/else statements)
- Functions
    - Move
    - Wait
    - Math
    - Number-to-Text
- Data types
    - Number
    - Text
    - Logic
- Sensors
    - Light
    - Sound
    - Ultrasonic
    - Display
    - Touch

Instructional Method

This instruction follows Dave Merrill's first principles of instruction. Adhering to these principles provides a structure for presenting the lessons in ways that help students to easily understand what to do and to succeed in learning the desired concepts and practices. The basic notion behind this approach is outlined in the following diagram:



[1]

Problem-based:

Notice that all principles are centered around a specific problem. This enables students to learn desired concepts while seeing and understanding their application in an authentic context. The lessons in this packet are designed to each get at a specific programming principle or computational thinking process in learning to program an NXT robot. We do this by using a central activity in each lesson around which we can discuss and apply the to-be-learned concepts and practices. Some of these activities require robots and others are "unplugged."

- ***Activation***: Build upon experiences people already know.
- ***Demonstration***: Show students what they will learn.
- ***Application***: Give students an opportunity to use what they've learned.
- ***Integration***: Encourage students to demonstrate, adapt, modify or transform what they've learned back into their own lives.

[1] Image retrieved from: http://edutechwiki.unige.ch/en/First_principles_of_instruction

Curriculum Map

This Teacher's Guide and the accompanying Student Workbook are comprised of several basic robots. Each lesson is designed to teach a new concept or tool important to Lego Mindstorms programming. To reinforce the new material additional challenge robots can be inserted in between the lessons that implement the new concept or tool. We decided on **three different teaching methods** for the instructions for the robots.

A. ***No Guidance***: Students are given the goal of the robot. They then plan out the program, this can be done on their own or with help from the instructor. Finally they program. This approach helps reinforce students' ability to break a problem into parts and create a plan.

B. ***Step by Step***: All the steps are outlined in the workbook, the students just follow the instructions step by step. We use this for more complex problems. Using this method, the student workbook poses several specific questions to the students to gauge understanding.

C. ***Final Picture***: The students are given the final picture of the program, which they copy in their programming. Although a lot can be seen in the picture, not all of the variables values are deducible from the picture alone, the students will have to think through the program. This method helps to reinforce students' ability to analyze a program, automate and interpret a plan.

Robot	Teaching Method
Human Robot	A
Constraints	A
Move	A
Touch Go	B
Line Following	B/C
Scaredy Bot	A
Light Reading	C
Touch Count	B/C
Back and Forth	C

Lesson #1 — Introduction to Functions & Robots

Goal: Students will program a person to complete a simple task (e.g., sitting in a chair on the other side of the room).

Learning Outcomes:

- Students will understand that a function is composed of an *input — process — output*.
- Students will practice programming a simple human robot
- Students will debug their robots

Principles Reinforced: functions, debugging

Time	Activity	Purpose	Resources
10 min	**Introductions**: Gather students in a circle. Each students presents him/herself and an action of something s/he likes to do. After everyone has gone, ask people to start with themselves (say name, do action), then "pass" the baton to another person by saying his/her name and doing the associated action	Learn names. Present idea that functions occur when called by name.	Candy (to encourage participation)
10 min	**Rules**: Ask students what rules you should have as a club. Work as a group to generate reasonable rules and expectations for everyone's safety and learning.	To establish order and shared management	Clipboard "Club Rules" worksheet
10 min	**Find a function**: Tell students that we are going to use lots of functions in this club. Functions are like a candy machine. Tell students that functions are made of 3 parts: an input, a process, and an output. Line the students up in a straight line and give them each a coin (or whatever your machine accepts). Tell them to each put the coin in the machine and turn the handle. They get their candy and sit down. Ask what the input, process, and output are. After discussion, send the kids to find an example of a function in the school. Tell them to not go beyond the hallway outside the classroom. Students share examples upon return.	To recognize that we are surrounded by functions	Sticky notes Pen or pencil A candy "machine." You can use yourself as the machine if you don't have one.
25 min	**GOAL**: Tell students they'll program a very simple robot (you). They can only give that robot simple commands, such as "go forward X steps" "turn left/right X degrees" and "sit." Groups will "program" their robot by creating a plan to get the robot to do something (e.g., sit in a chair somewhere down the room or hallway; take a drink from a drinking fountain; etc.) Execute the program EXACTLY as students give it to you. **Vocab**: emphasize that when students are trying to figure out where their program went wrong they are *debugging*. *Modification*: if students finish early, ask students how to make the program more efficient. Is there some "variable" they could change?	Students will learn that a robot does exactly what you tell it in the order you tell it.	Blank paper pencils

Lesson #2 — Programming with constraints

Goal: (1) To make the robot start at a line, go to another, and return, by programming only using the on-board commands. (2) To do a similar exercise using the NXT software and the MOVE function.

Learning Outcomes:

- Students will be able to program a robot on the Lego brick
- Students will be able to program a robot using the Lego Mindstorms NXT software, using only a single function
- Students will recognize the different variables available to the "Move" function
- Students will identify their goal and plan before programming.

Principles Reinforced: Constraints, Variables, Goals, Planning

Time	Activity	Reason	Resources
5 min	Ask students questions about prior activity. Ask questions about functions and debugging.	Activate prior knowledge	Candy? (could do this without)
15 min	**Demonstrate**: Put a robot on the ground in front of 1 piece of tape and tell students to gather around. Press the "go" button to get the robot to proceed to the next piece of tape. Tell students that we will begin programming the robots, but that they can only use the on-brick program. These restrictions are called *constraints*. Explain that all programs have constraints and that we'll deal with them repeatedly throughout our experience. **Activate**: After the demonstration, ask groups to write down the goal. We will start all programming activities by writing out our goal. Then ask students to create a plan for how to get the robot to achieve that goal in the 5 steps allowed on the brick. **Modification**: Ask students to get the robot to go to the second piece of tape and back (tape will have to be about 5' apart or less for this to work).	Anticipatory set Plan and program within constraints	• NXT robot with pre-programmed sequence • NXT robot for each student • tape • planning sheets (blank paper?)
3 min	**Demonstrate**: Call students together and demonstrate a robot that will: (a) start out at a line, (b) go to a second line, (c) turn 180 degrees, (d) return to the first line, and (e) turn 180 degrees (essentially returning the robot to its starting point). Tell students this is their next goal and ask groups to write down that goal on a new planning page.	Anticipatory set	• NXT robot with pre-installed "there and back" program • planning page
10 min	Show students the NXT software. Indicate that we'll only use a single function, the "MOVE" function, to achieve our program. This is a *constraint*. Show students the properties of the move function. Note that these are *variables*. Review the different variables of the MOVE function and their possible values.	• Understand that functions rely on variables. • Understand that the value of a variable changes, but the name never	• projector • computer with NXT software

		does.	
25 min	Groups plan and program their robots for use with the computer software. Make sure students show you their plan before they program the robot. The plan doesn't have to be perfect, but they need to think through it first.	Understand how different variables affect robots' performance.	• NXT software installed on each group computer • NXT robots

Lesson #3 — Move! (Sensors and Switches)

Goal: Create a robot that moves when you press a button and stops when you let go.

Learning Outcomes:

- Students will understand conditional logic
- Students will know how to test sensors using only the brick

Principles Reinforced: Conditions, Switches, Loops, Data analysis

Time	Activity	Reason	Resources
5 min	Review prior activities. Show students NXT software and "MOVE" function. Ask students what the properties at the bottom are called (*variables*). Ask what each of the variables means.	Activate prior knowledge	• NXT software • LCD Projector • Computer with NXT software
5 min	**Light reading demonstration**: Gather students in a horseshoe so they can see the menu of an NXT robot as you operate it. Navigate to the sensor testing screen and test the reflected light.	Demonstrate how to test sensors	• NXT robot with connected light/color sensor
10 min	Ask students to **gather data from several different surfaces**, light and dark, in the room and write that down on their light worksheets. After all groups get 10 readings, ask them what those readings mean.	Gather and analyze data	• Reflected light worksheets • Group NXT robots
10 min	**Switch Game**: Tell students you're a train track operator and they are a train. They are to tell you a number between 1-100 and you send them to one side or the other based on a pre-determined condition (e.g., choose the number 50). After students are sorted, ask them to guess your condition.	Students experience conditional logic with a concrete, familiar example.	
5 min	Introduce SWITCH function on the projected screen. Show students the variables associate with the switch. Tell them this is like the switch operator in the game you just played. They will need this function to create a controller for their robot.	Introduce new function.	• NXT software • LCD Projector • Computer with NXT software
20 min	**GOAL**: Students will program the NXT robot to advance forward when the user is pressing a button down (touch sensor) and will stop when the button is not pressed. MAKE SURE students write down their goals and plans before programming. *Modification*: Challenge students to figure out what other things their robot can do using only a switch, a single touch sensor, a loop, and move function.	Students will practice creating a condition for the robot to respond to.	• planning sheets • group NXT robots • installed NXT software

Lesson #4 — Line-following Robot

Goal: Program a robot to follow a black line on the floor.

Learning Outcomes:

- Students will plan a program involving 4 functions (switch, sensor, move, loop)
- Students will program a robot to follow a solid line on a light surface.

Principles Reinforced: Debugging, planning, conditional logic, loops, optimization

Time	Activity	Reason	Resources
5 min	Review prior activity. Ask about conditions and loops.	Activate prior knowledge	Candy?
5 min	Explain that today groups will combine what they learned previously to create a robot that will drive itself automatically, similar to how airplanes follow a path automatically. Place a robot on the ground in front of a zig-zag line of tape (about 6 feet long). Ask students to observe how the robot moves as it follows the tape. Discuss this movement as it will be important for them to understand that the robot never moves both motors at the same speed at the same time.	Anticipatory set	• tape • NXT robot with working line-following program
10 min	**Planning demonstration**: Inform students that their goal is to get their robots to follow the line as fast as possible using only the functions they've already learned (switch, move, loop, light sensor)	Model how to think through a more involved plan	• planning page • student attention (very important this time)
20-30 min	**Programming**: Allow students to plan and program their robots and test on the line. It will be useful to have 2 lines so that there is not too much congestion during testing. Remember that students must show you their written goals and plans before programming.	Plan a robot that follows a more concrete path	• Group NXT robots • Installed NXT software
10-20 min (overlap w/ above)	**Optimization**: Once a group gets their robot to follow a line, challenge them to make it complete the run faster. Have students fill out their Line-following worksheets, identifying the variable(s) they change each time and the result.	Understand how slight variation in variables can make a big difference in execution	• Line-following Course Worksheet

Lesson #5 — "Scaredy-bot"

Goal: To create a robot that waits for a loud noise, gets "scared" and then runs around the room continuously without running into anything, saying something each time it gets too close.

Learning Outcomes:

- Students will know how to use the *sound* and *ultrasonic* sensors, as well as the on-board speaker.
-

Principles Reinforced: Debugging, Planning, Conditions, Switches, Loops, Optimization

Time	Activity	Reason	Resources
Before class starts	Begin by sitting and asking them what sort of things scare or startle them (Get ideas from everyone to get all people participating if possible). Ask what was scary about it. You can use either a student's example or your own (in my case, a scary movie) and talk about how sudden loud noises can scare someone. Sometimes we get scared by something we see. Explain that today we are going to program our robots to be "scared" of a loud noise we make and when they see something. This activity should be started before class starts to help keep students' focus when the lesson begins.	To help keep class members focused	• Candy
5 min	Review prior activities. Ask students what is the input on a robot (*sensors*). Ask what types of sensors they know.	Activate prior knowledge	• NXT software • LCD Projector • Computer with NXT software
10 min	**Scaredy-bot demonstration**: Gather students in a circle so they can see the robot move. Remind the class to be quiet and have a student volunteer activate the program and make a loud noise (if their initial noise doesn't cause a reaction, tell them to get closer to the sound sensor). Have the class observe the robot's reactions.	Demonstrate how the program should function	• NXT robot with connected sound and ultrasonic sensor • Scaredy-bot program should be installed
20-30 min	**Programming**: Allow students to plan and program their robots and test out the sensors and the robots "scared" reactions. Remember that students must show you their written goals and plans before programming.	Plan for the robot to react to a noise and nearby objects.	• An open space where the robot won't react too quickly using the ultrasonic sensor.
10-20 min	**Optimization**: Once a group succeeds in the task, have them complete a challenge: 1. If there are only about 10 minutes left, students can modify their program to have their robot end	Allows students to critically think about how to	• Scaredy-bot Course Worksheet

	with a stimulus of another noise. 2. If more time is available (~10+ minutes), they can program their robot to run until it finds a dark place (under a table) and “hides” there. This would mean the inclusion of a light sensor and measuring the ambient light (or reflective light, if they want) of the room and under the table and taking an average to program into their robot.	change the program for different parameters.	

Lesson #6 — Light Reading Program

Goal: Create a program that reads light values and displays them on screen

Learning Outcomes:

- Students will build a light-reading program in NXT.
- Students will be introduced to the Number-to-Text and Display functions and how they work together.

Principles Reinforced: Debugging, Planning, Conditions, Switches, Loops, Optimization

Time	Activity	Reason	Resources
5 min	Review prior activities. Remind students of when they did Light Reading on the LEGO brick itself. Ask them what the parts are for that program (*light sensor, display, and loop should be what they can guess now*)	Activate prior knowledge	• NXT software • LCD Projector • Computer with NXT software
10 min	**Light Reading Instruction**: Have students observe on the projected screen or your computer screen the program for Light-Reading as you create it. Walk through the logic with them. Explain what the Number-to-Text function is.	Demonstrate how the program should be programmed	• NXT robot with connected sound and ultrasonic sensor • Scaredy-bot program should be installed
5-10 min	**Light Reading Demonstration**: Demonstrate the Light Reading program after you have successfully built it. Ask how it differs from the light reading function they did before on the brick. If needs be, demonstrate both (*The secret is the wait timer determines how long it will display the measured light before taking a new measurement. Ask how this could be useful or not useful*).	Demonstrate how the program works compared to the LEGO brick function	•2 NXT robots, 1 with the new program and 1 using the LEGO brick function
15-20 min	**Programming**: Allow students to plan and program their robots. Remember that students must show you their written goals and plans before programming. Have them play around afterwards with the Wait-Timer function to determine what they feel is the best time.	Plan for the robot to be able to measure reflective light	• Different colors and reflective surfaces to test the sensor on
10 min	**Discussion**: Once each group has completed the task (it shouldn't take them too long with directions) you can wrap up with discussion about what ways Light-Reading could be used in real-life settings (*self-driving cars, automated shutters, self-adjusting solar panels, etc.*). Also ask what they felt was the best time for the Wait-Timer function and why.	Allows students to critically think about how to change the program for different parameters.	• Light-Reading Course Worksheet

Lesson # 7 — Touch Count

Goal: Program a robot to count how many times a button is pushed in a limited timeframe and show that result on the display.

Learning Outcomes:

- Students will learn how to create their own variable
- Students will review the input – process – output of a function through the math function

Principles Reinforced: functions, variable

Time	Activity	Purpose	Resources
10 min	**Review**: Review the different data types introduced in the previous lesson (number, text, logic). In addition, go over the math function and what the input, process, and output of it are.	Review data types and functions.	Candy (to encourage participation)
5 min	**Demonstrate**: Place the robot on the table and ask them to guess how many times you can push the button in 10 seconds. Mention that at the end there will be a competition to see who can push the button the most in 10 s. Also introduce the idea that they will be creating their own variable (as opposed to one used by an existing function)	To provide the end goal.	Robot with a working program
30 min	**Programming**: Have the students work from their packet to plan and program the robots. Go around to each group and have them step through the program and explain how it works. When they finish have them report the highest number of times they could push the touch sensor in 10s.	Program and demonstrate understanding.	Robots with the touch sensor attached. A place to record the scores, like a white board.
10 min	**Challenge:** If students are done have them complete the additional challenges outlined in the packet.	To have the students apply what they learned	
5 min	**Clean Up**		

Lesson #8 — Back and Forth

Goal: Program a robot to go back and forth between two pieces of tape (about 10-16" apart) as many times as possible in 20 seconds.

Learning Outcomes:

- Students will apply everything learned in prior lessons in a single robot
- Students will create custom variables (to store direction)
- Students will convert numbers to text and display the value on screen

Principles Reinforced: logic variable, switch, loop, wait

Time	Activity	Reason	Resources
5 min	**Review:** Ask the students to explain different data types (number, text, logic). Specifically go over logic variables.	Remind them of logic variables.	
5 min	**Demonstration:** Place the robot on the table and ask students to predict how many times it can travel back and forth between the two lines in 20 seconds. Once all guesses are given, start the robot and cheer it on. Check students' answers against the actual count.	Anticipatory set	• Robot with Back and Forth program, • Two lines of dark tape on a flat, lighter surface, about 10-16" apart.
30 min	**Programming:** Have the students work from their packet to program their robots. (Students are able to program almost the entire robot by looking at the completed program. One thing they may struggle with is creating a logic variable to keep track of which direction the robot is travelling in) Visit each group and have them explain the program as outlined in their packets. **Challenge**: See who can get the their robot to go back and forth the most times in 20 seconds. (e.g., ask what variable they could change to make it faster) **Common problems** include not checking the reflected light of the floor/ tape and adjusting the wait function accordingly and not setting the variables to true/false.	To have the students apply what they already learned Gauge students' ability to read a completed program.	• Robots with light sensors • Maybe multiple courses with the same distances between the lines of tape
10 min	**Challenge:** Have the students add to the program so that it counts and displays how many times it goes back and forth (HINT: they did this in the touch count robot)	Application of knowledge to a new situation	
10 min	**Clean Up**		