EdScratch lesson plans

Parent's guide and answer key for unit: *Move it!*

Document number: 3.2.4.3.1 Rev 1.1

Contents

Contact us

Have any questions regarding these lessons? Our team are here to help.
Please feel free to contact us at
https://meetedison.com/edison-robot-support/contact-us/

About this guide

This guide offers parent overviews, delivery recommendations and other supporting information for the EdScratch home lesson activities available at https://meetedison.com/learn-at-home-learn-to-code-with-edscratch/.

These lessons have been selected from the full EdScratch activities set (available at https://meetedison.com/content/EdScratch/EdScratch-student-lesson-activities.pdf) and are designed to allow students to work independently at home. As students work through the activities they will develop a familiarity with computational thinking and fundamental computer science concepts, mastering a range of related skills as they progress. This supplemental guide outlines more about the concepts to be taught, approaches to be taken, skills to develop and processes to be learned. Here you will find strategies, ideas, and further information to help make teaching using Edison and EdScratch at home effective, easy and fun.

This guide provides an overview and supporting information for all of the home EdScratch lessons and activities. In this guide you will find:

- an overview of each lesson
- a dedicated section per activity detailing:
  - the activity's purpose and objectives,
  - supporting information including estimated time requirements, supply requirements, and tips for delivery, and
  - an answer key including recommendations on assessing student work.

Creative Commons licence

These teaching recourses have been released under a Creative Commons licence. You are free to use these resources as they are, translate them, share them or use them as the base to develop your own customised lessons.

Licence and attribution details

Developed and written by: Kat Kennewell
Illustrations by: Jin Peng

EdScratch is developed by Microbric Pty Ltd using open source software created and maintained by the Scratch Foundation. The Scratch Foundation does not sponsor, endorse, or authorize this content. See scratch.mit.edu for more information.

[1] Creative Commons Attribution – ShareAlike 4.0 International License - http://creativecommons.org/licenses/by-sa/4.0/

How to use this guide

Understanding the activity types

The EdScratch lessons are comprised of activities designed to gradually introduce and reinforce computational thinking, computer science and problem-solving skills. All lessons include base activities which introduce key ideas and contain tasks requiring students to apply what they have learned. Supplemental extension activities, designed to further explore and expand on the core learnings of each lesson, are also included.

There are three types of lesson activities: ‘Let’s explore’ activities, ‘Bonus challenge’ activities, and ‘Skill challenge’ projects.

Let’s explore activities – core learning

These activities are the base activities which introduce new concepts and explain key ideas. ‘Let’s explore’ activities include explanations and guidance around the core learnings, then have tasks for students to complete. The tasks allow students to explore the concepts and skills being introduced.

Bonus challenge – extension activities

‘Bonus challenge’ activities are extension activities which expand on and reinforce the material that has been previously introduced. The tasks and projects outlined in a ‘Bonus challenge’ activity are generally more open-ended and offer less structured guidance than ‘Let’s explore’ activities but still include sufficient guidance and hints to help students undertake the tasks or projects.

Skill challenge projects – extension activities

Similar to ‘Bonus challenge activities, ‘Skill challenge’ projects are also extension activities which act as ‘summary’ challenges, reinforcing the material that has been previously introduced. The tasks and projects outlined in ‘Skill challenge’ activities are generally more open-ended and offer less structured guidance than ‘Let’s explore’ activities and will take students longer to complete.

Reading the activity overview

Every EdScratch lesson activity has a dedicated section included in this guide. Per activity you will find the following information:

- the activity’s name,
- an estimate of the activity’s size,
- delivery recommendations (if any),

- a list of the supplies and resources needed for the activity outside of the lesson activity,
- an overview of the activity,
- any tips or tricks to help in delivering the activity, and
- an answer key with assessment suggestions (as needed).

Using the answer key

All of the EdScratch activities have tasks or projects to be completed. Many of the worksheets also require students to note down elements of their experience or to provide answers to questions in various formats. For every activity which has students provide answers to a question or questions, you will find an answer key in that activity's section.

The answer key lists both the question number and question type, then provides the answer (or sample answer) along with marking notes, as required.

Example answer key:

Question	Type	Sample answer	Marking notes
1	SE	*When I turned Edison on, the red LEDs came on and started flashing. Edison also made a chirping noise one time.*	There are two key things to look for in students' answers: - The robot's red LED lights turn on and begin to flash. - The robot makes a noise.

Answer types

The answer type column identifies what style of response the question has asked students to provide. There are three possible answer types:

- **Exact answer (EA)**: a question which has an exact solution which the student must provide.
- **Result code (RC)**: a question which asks students to capture the code they have as a result of their programming, where a variety of code solutions are possible.
- **Student experience (SE)**: a question which asks students to capture their experiences or outcomes as a result of their experimentation and programming, where a variety of responses are possible.

It is valuable to note the answer type when marking students' work. When marking result code (RC) or student experience (SE) type answers, it is important to remember that many answers are possible.

Supplies you will need

All lesson activities require a minimum of one V2.0 Edison robots and internet-connected programming devices (desktops, laptops, Chromebooks or tablets).

If you and your student are new to Edison robots or EdScratch, it is strongly recommended that you complete the following prior to this unit:

- Read and follow the instructions for setting up your programming device covered in the teaching guide for lesson 1: https://meetedison.com/content/Hour-of-Code/EdScratch-and-Edison-getting-started-teacher-guide.pdf
- Have your student complete lesson 1: https://meetedison.com/content/Hour-of-Code/EdScratch-and-Edison-getting-started-student-guide.pdf
- Have your student complete lesson 2: https://meetedison.com/content/Edison-at-home/Lesson-2-warning-messages.pdf[2]

Students should then be prepared to work through this unit independently.

The majority of the 'Let's explore' base activities can be completed using just basic supplies (see list below) along with the student worksheets and activity sheets included in the EdScratch lesson activities set. Many of the 'Bonus challenge' and 'Skill challenge' extension activities require additional supplies to enable students to get the most out of physical computing. The following lists detail the basic supplies and key additional resources you need to get the most out of the EdScratch lesson activities.

Basic supplies:

- Edison robot and EdComm programming cable
- Programming device (computer or tablet)
- 4x AAA batteries per robot
- Print-outs of student worksheets and activity sheets

Additional resources (specifics are noted per lesson):

- Dark coloured tape (e.g. black electrical tape)
- Torch (flashlight)
- Opaque objects to use as obstacles and walls for 3D mazes
- TV or DVD remote control
- EdCreate kit or any other LEGO brick compatible building system parts
- Various 'maker-space' craft and build materials [such as coloured construction paper, large-sized paper (e.g. butcher's paper), glue, felt, cardboard, pipe cleaners, recycled materials, and other similar materials]

[2] Parent's guide for lesson 2 - https://meetedison.com/content/Edison-at-home/Lesson-2-Parent-guide.pdf

Frequently asked questions

The following section includes some of the most common questions about using the EdScratch home lesson activities set. Additional information about using the Edison robots, including troubleshooting help, is available at the Meet Edison website: www.meetedison.com

Q: How much do students need to already know about coding and robotics to use these lessons?

A: No prior knowledge of either coding or robotics is needed to complete the EdScratch home lessons. We recommend working through the home lessons in the order they are presented to be the best way to build up your knowledge and get the most out of these activities.

Q: Do students need to know Scratch to use these lessons? Do these lessons tie into Scratch?

A: The lessons assume that students have not used a Scratch programming environment previously. If students are familiar with Scratch, however, they will likely find using the EdScratch environment fairly straightforward.

The EdScratch lessons cover core coding and computer science concepts that can also be taught or reinforced using MIT's Scratch programming environment. However, EdScratch and Scratch are separate programming languages – programs written in one language are not compatible with the other.

Q: Are the solutions in the answer key the only correct answers?

A: In some cases, there is a single correct answer, but in most cases many solutions are possible. Programming is creative – don't be too quick to dismiss your students' ways of thinking. What might appear to be 'incorrect' can be an acceptable solution, even when a more elegant solution is also possible.

Q: Can I modify the suggested activities and projects?

A: Absolutely! Tools, such as Edison and EdScratch, can be powerful motivators on their own. However, to get the most out of any 'cool tool', it needs to relate directly to the interests of the students using it. Choosing extension activities and projects that are the most relevant to your students' lives and interests will help them to get the most out of the lessons overall.

Q: What reading level do students need to read at to use the student worksheets independently?

A: The original set of student worksheets, written in Australian English, are designed for independent use by students in Year 5 (10-11 years old) and above. Overall, the complete worksheet set has an average reading level of 6.6 on the Flesch-Kincaid Grade Level scale with later units averaging slightly higher reading requirements than earlier units.

Please note: Translated and modified versions of these lessons may have different reading level requirements.

Q: I have a Version 1 Edison robot. Can I use EdScratch and these lessons?

A: As a general rule, you can use Version 1 Edison robots which have the latest firmware update (available at https://meetedison.com/edison-robot-support/firmware-update/) with EdScratch and the activities in this EdScratch home lesson set. However, there are a few limitations, especially when it comes to Edison's motor outputs. When using EdScratch blocks from the 'Drive' category in EdScratch with Version 1 robots, only use 'seconds' as the distance units input parameter. Using 'cm', 'inch', or 'degrees' as the distance units input parameter will not work with Version 1 robots and can result in some bizarre behaviour from the robot. In all activies and projects where students use 'Drive' category blocks, be sure to remind students to always and only use 'seconds' as the distance units input parameter.

It is also recommended that you avoid using the sensing and event EdScratch blocks related to 'drive strain' with Version 1 robots as these blocks can cause the robots to behave erratically. (*Note:* none of the activities in these lessons use the 'drive strain' sensing or event blocks.)

What's the reason behind these limitations? The key cause has to do with some physical differences between Version 1 Edison robots and V2.0 Edison robots. Edison V2.0 robots have wheel encoders which allow the robots to travel specific distances at exact speeds. These encoders enable the robots to use distance units other than time, including centimetres, inches and degrees. Version 1 Edison robots do not have encoders and therefore are not capable of the precision driving required to use these other inputs.

Lessons overviews and answer keys

Lesson 3: Let’s explore how computers ‘think’

Estimated time	30-45 minutes
Additional resources needed	nil

Overview

In this offline activity, students are introduced to the ideas of computational thinking and sequence. Students do not need to fully understand what computational thinking entails at this point, but it is valuable to use the term explicitly. Using computational thinking makes coding much easier, but it is a bit different from how we normally approach and think about things in our lives. That’s why introducing ‘computational thinking’ as a way to think about things is a useful notion. Getting kids to ‘put on their computational thinking hats’ can help them approach problems differently, finding solutions they may otherwise overlook. This activity has students practice following and giving good sequential instructions by being explicit about each action and going step-by-step, one action at a time. Developing a sequential approach to problem-solving is crucial for good computational thinking and success when programming in EdScratch.

Answer key

Question	**Type**	**Answer/Sample answer**	**Marking notes**
1	*EA*	*The car.*	
2	**EA**	The car.	
3	**EA**	*The ice cream cone.*	
4	**SE**	Move forwards 3 squares. Turn left. Move forwards 2 squares.	Any answer which meets the question’s criteria is acceptable.
5	**SE**	Move forwards 2 squares. Turn left. Move forwards 2 squares. Turn right. Move forwards 1 square.	Any answer which meets all of the question’s criteria is acceptable.
6	**SE**	Turn right. Move backwards 1 square. Turn right. Move backwards 1 square. Turn right. Move backwards 1 square.	Any answer which meets all of the question’s criteria is acceptable.

Lesson 4: Let’s explore going step-by-step in EdScratch

Estimated time	15-30 minutes
Additional resources needed	nil

Overview

This activity gets students to apply the idea of sequence and sequential programming to the EdScratch environment. Exploring how changing the sequence of the blocks on the screen affects the robot’s behaviour helps solidify the importance of sequence in programming. The idea that each EdScratch block is one ‘action’ is introduced. Understanding this idea will help students apply sequence when building programs. Students also experiment moving blocks around and changing inputs inside blocks to gain confidence manipulating the EdScratch language.

Tips and tricks

- If students are struggling to get blocks to move from the block pallet, make sure that they are dragging the blocks straight to the right into the programming area, not in another direction (such as up or down inside the blocks pallet itself).
- Blocks must be attached to the start block to be downloaded to Edison. Floating blocks not attached to a start block will not download to Edison.
- You can remove blocks you don’t want by dragging them into the trash bin in the lower right corner of the programming areas or back into the block pallet.
- While the term ‘input parameter’ is not introduced until the next activity, beginning to refer to the options inside a block as ‘input parameters’ is a habit you may want to get into straight away.

Answer key

Question	Type	Answer/Sample answer	Marking notes
1	EA	First, the robot will drive forwards for 2.5 seconds at speed 5. Second, it will beep. Third, it will turn the left LED on. Fourth, it will spin left for 120 degrees at speed 10. Fifth, it will drive backwards for 2 seconds at speed 2.	NOTE: Task 2 encourages kids to find and fix any errors in what they have written in either this answer or their program.
2	SE	First, the robot will drive forwards for 2.5 seconds at speed 5. Second, it will drive backwards for 2 seconds at speed 2. Third, it will spin left for 120 degrees at speed 10. Fourth, it will turn the left LED on. Fifth, it will beep.	Student answers should only move the existing blocks around – not add or change blocks.

Lesson 5: Let's explore driving Edison

Estimated time	30-45 minutes
Additional resources needed	Optional: supplies for making your own tracks

Overview

Get the robots moving by applying sequential coding to the 'drive' category of blocks in EdScratch. Kids work through two programming tasks to get the robots to complete driving tasks – first, a straight track and then a mini maze. The mini-maze task requires students to use computational thinking to create a sequential program to solve the maze. To complete the maze without 'cheating' students inevitably need to iterate different versions of their initial program concept, experimenting with different blocks and input parameters.

Tips and tricks

- You can also have students make their own straight tracks or mazes to complete on large paper or by using tape to mark out a track on the floor or desk.
- First attempts in programming are almost always 'wrong' – and that is perfectly okay! Persevering and changing programs iteratively, testing as they go, will get students to the right answer. Resilience is key!
- Due to minor mechanical differences in the motors and encoders inside different Edison robots, some robots may not turn to exactly 90 degrees when given the input of 90. Encourage students to try different values around 90 (e.g. 87 or 93) to find the input that works best for their Edison.

Answer key

Question	**Type**	**Answer/Sample answer**	**Marking notes**
1	RC	Forwards for 26.5 cm at speed 5 OR forwards for 10.5 inches at speed 5 OR forwards for 1.05 seconds at speed 5	Student answers will depend on the input parameters they select. Sample answers assume completion on the provided activity sheet using speed 5.
2	**SE**	1. Go straight. 2. Turn left. 3. Go straight. 4. Turn left. 5. Go straight. 6. Turn right. 7. Go straight then stop.	Students may miss steps in their initial plan – creating the program will show them any issues.

Lesson 5 - BONUS Challenge: Maze madness

Estimated time	15-45 minutes (depending on the number of tasks attempted)
Additional resources needed	Optional: supplies for making your own tracks

Overview

Perfect for students who finish the previous activity quickly or for students needing additional task-based sequential programming practice, this set of 3 extension ideas offers variations on the maze-driving task.

Tips and tricks

- If students are making their own mazes to complete, they will be most successful doing so on large paper or by using tape to mark out a track on the floor or desk.
- Due to minor mechanical differences in the motors and encoders inside different Edison robots, some robots may not turn to exactly 90 degrees when given the input of 90. Encourage students to try different values around 90 (e.g. 87 or 93) to find the input that works best for their Edison.
- Encourage students to try both the ‘turn’ and ‘spin’ input parameters to see which works best.

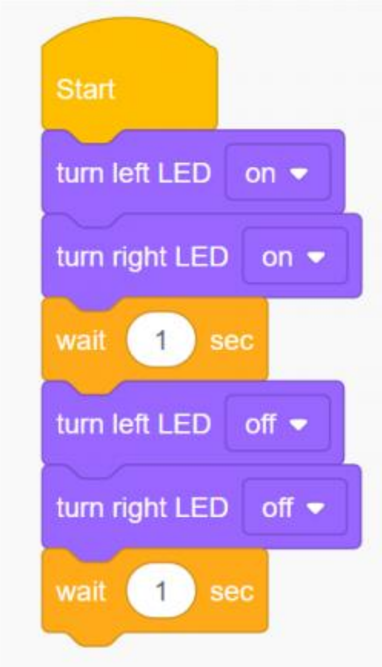
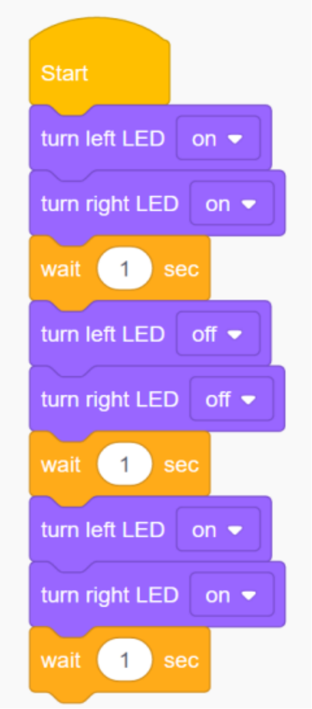
Lesson 6: Let’s explore Edison’s outputs

Estimated time	30-60 minutes
Additional resources needed	nil

Overview

The input-process-output cycle is introduced and then experimented with in this activity with an emphasis on observing the output results of sequential programming. Students begin to explore the ‘LED’ and ‘Sound’ categories of blocks in EdScratch and are introduced to their first control structure – the ‘wait’ block. Beginning to understand that the input-process-output cycle is at play in every program helps students start to decompose programming into separate stages. This is an important skill for debugging in coding and for the computational thinking skill of decomposing problems into smaller chunks.

Answer key

Question	Type	Answer/Sample answer	Marking notes
1	EA	- Turn Edison's lights off -- LEDs - Play a musical note -- Sound - Spin Edison right -- Drive	
2	SE	The robot did all the steps, but it was really fast, so it sort of seemed like it wasn't doing them in order but all at once.	Students may note that the execution of the outputs is very quick.
3	RC	 Start turn left LED on turn right LED on wait 1 sec turn left LED off turn right LED off wait 1 sec	What works 'best' is subjective, but students should have a wait block in between the LEDs being turned on and being turned off at a minimum.
Mini-Challenge	N/A	 Start turn left LED on turn right LED on wait 1 sec turn left LED off turn right LED off wait 1 sec turn left LED on turn right LED on wait 1 sec	The worksheet does not instruct students to share their answer, but you may choose to have students present their thinking or resulting code.

Lesson 6 - BONUS Challenge: Drive the maze safely

Estimated time	15-30 minutes
Additional resources needed	nil

Overview

Overview Students will need to put their understanding of sequence into careful practice in this activity. Writing a program that has the robot perform all the steps to complete the maze plus indicating turning direction using the LED lights, the buzzer or both before making each direction change is a great way to let students apply their understanding of sequence in a more creative programming challenge.

Tips and tricks

- Remind students that they can use 'wait' blocks to have the robot pause for a set amount of time.
- While the worksheet doesn't ask students to capture their program, you may choose to have students write down or demonstrate their program and explain their reasoning for using the blocks, inputs, and sequence they chose.
- Due to minor mechanical differences in the motors and encoders inside different Edison robots, some robots may not turn to exactly 90 degrees when given the input of 90. Encourage students to try different values around 90 (e.g. 87 or 93) to find the input that works best for their Edison. • Encourage students to try both the 'turn' and 'spin' input parameters to see which works best.

Lesson 7: Let's explore input parameters

Estimated time	15-30 minutes
Additional resources needed	nil

Overview

This offline activity explores the concept of input parameters, building on the basic definition introduced in an earlier student activity. This activity helps students understand the concept of input parameters and establishes a framework for working out what any given input parameter is contributing to a program. Being able to think critically about input parameters helps students to understand what is actually happening in a program. Input parameters contribute critical information to the inputs of programs. By explicitly examining input parameters, students gain a deeper understanding of the inputs that they are using in their programs. The ability to link the input components of any given program to the expected outputs of that program is an important skill to develop for computational thinking and problem-solving.

Tips and tricks

- You may want students to use the programs in the worksheet with their Edison robots to see the programs in action, helping to cement the connection between inputs and outputs.
- To further demonstrate the connection between input parameters' effects on outputs, have students adjust the input conditions of the worksheet programs in EdScratch, hypothesise on what those changes to the inputs will do to the outputs, and then test the programs with their robots to see the results.

Answer key

Question	Type	Answer/Sample answer	Marking notes
1	EA	Seconds	
2	EA	7	
3	EA	Drive backwards for 6.5 seconds at speed 7.	
4	EA	4	While this program has multiple blocks, only one of the blocks has input parameters
5	EA	What direction do you want the robot to go?	Any answer that shows an understanding that this input parameter is related to direction is acceptable.

Lesson 8: Teach Edison to count to 9

Estimated time	30-45 minutes
Additional resources needed	nil

Overview

This activity offers programming challenges which put some of the skills students have been developing to use. Creating programs for the Edison robot to ‘trace’ (i.e. drive in the pattern) the digital display numbers of their choice requires the application of sequential thinking and an understanding of the input-process-output cycle.

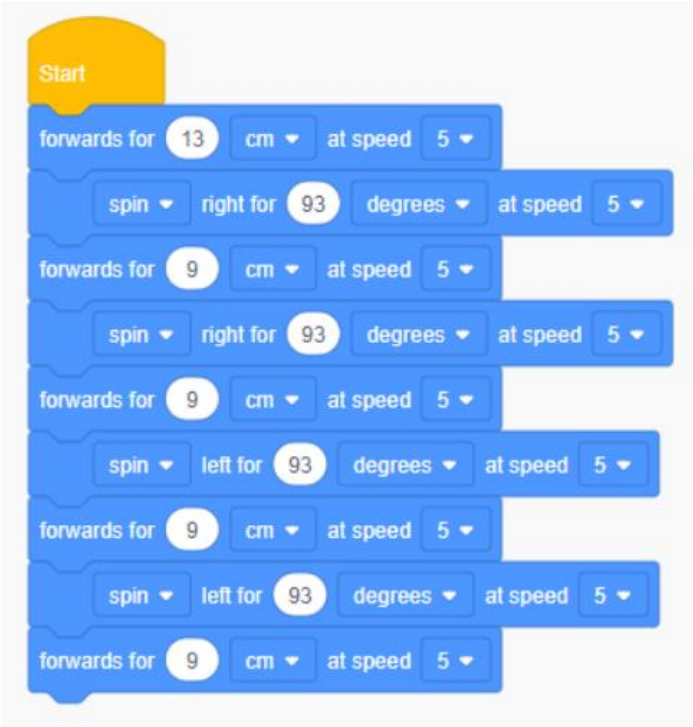
This activity calls attention to one particular input parameter (distance in the drive blocks) and asks students to identify patterns regarding the input parameter’s effects on output. Students then extrapolate from what they observe, applying their understanding to a different program challenge.

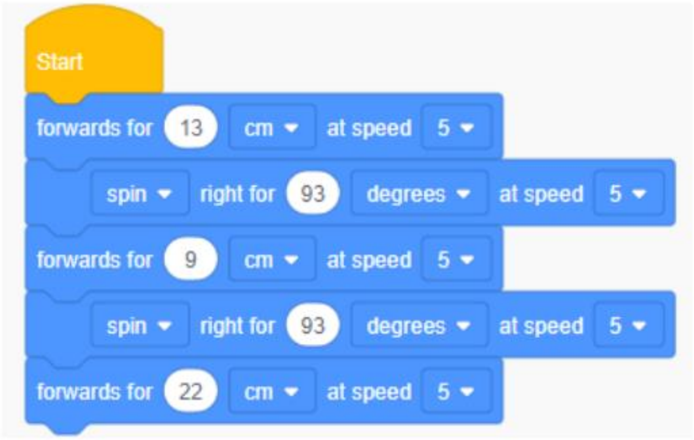
Tips and tricks

- Each segment in all of the digital display number activity sheets is the same length. However, when Edison turns, the robot does not turn on the spot but travels a bit. As a result, the distance parameter required to complete any segment after a turn will be smaller than the distance parameter needed to complete a straight segment.
- Due to minor mechanical differences in the motors and encoders inside different Edison robots, some robots may not turn to exactly 90 degrees when given the input of 90. Encourage students to try different values around 90 (e.g. 87 or 93) to find the input that works best for their Edison.
- Encourage students to try both the ‘turn’ and ‘spin’ input parameters to see which works best.
- It is possible to write a program for any of the numbers in the activity sheets without the robot needing to trace over the same segment twice. You may choose to mention this to help students envision the sequence they need to use in their programs.
- If Edison’s wheels catch the edge of the activity sheet’s paper, this can throw the robot off slightly. You can fix this by taping the activity sheet down or replicating the activity sheet’s pattern on a larger piece of paper.

Answer key

Question	Type	Answer/Sample answer	Marking notes
1	SE	2	

2	RC	 Start forwards for 13 cm at speed 5 spin right for 93 degrees at speed 5 forwards for 9 cm at speed 5 spin right for 93 degrees at speed 5 forwards for 9 cm at speed 5 spin left for 93 degrees at speed 5 forwards for 9 cm at speed 5 spin left for 93 degrees at speed 5 forwards for 9 cm at speed 5	
3	SE	The distance input parameter for the first drive block is slightly longer than all the others. All of the other blocks use the same number for the distance input parameter. I think this means that any digital display number will only need one of the two distance input parameters: one for a straight segment and one for after a turn.	Ideally, student answers will note a correlation between the value needed in an input parameter and the repeated real-world distance the robot needs to travel.
4	SE	7	

5	RC	 Start forwards for 13 cm at speed 5 spin right for 93 degrees at speed 5 forwards for 9 cm at speed 5 spin right for 93 degrees at speed 5 forwards for 22 cm at speed 5	You may choose to see if students used the same distance input parameters in each program.
6	SE	The distance input parameter for the first drive block, which is straight, is the same in both programs. Any drive blocks following a turn are shorter, but the same as each other. The distance input parameter in the turns is always the same. The last block of the program for ‘7’ is the same as the first drive block in each program plus the drive block after a turn because it is a combination of two segments.	Ideally, student answers will note a correlation between the value needed in an input parameter and the repeated real-world distance the robot needs to travel.

Lesson 8 - BONUS Challenge: Teach Edison to count to 9 out loud

Estimated time	15-30 minutes
Additional resources needed	Optional: supplies for making your own digital display numbers

Overview

Bring multiple outputs together into a single sequential program in this semi open ended challenge. There’s no real ‘right’ or ‘wrong’ way to complete this programming challenge. Any program that meets the criteria of ‘drive the number’ and ‘count the number’ is okay. This is an important concept for students to begin to understand.

Programming is highly creative, and there are often many solutions possible. That's part of what makes computer science so fun and interesting.

Tips and tricks

- Students can use the Sounds outputs, the LEDs outputs, both or a combination of the two to count. (They can probably also use the Drive outputs to count if they can figure out a way to do it while still tracing their display number.) For example, if the student has selected the number 5, their Edison might drive a segment of the display number, beep, turn, then drive the next segment, beep, turn, etc. until it has driven all segments and beeped five times. Alternatively, the program could drive the whole path, then flash the robot's left LED on and off five times. Any program where the robot drives the path and somehow signals the same value as that path is fine!
- If you want to highlight the idea that many different solutions are possible, split your students into groups for this activity and give each group the same digital display number. Have each group demonstrate their solution for the class. Discuss the differences and similarities in each group's program and highlight how, even if they are different, by meeting the criteria, they are still all correct.
- If you have students make their own numbers, you may want to give them a hint as to what segments they have to work with: a standard display grid is fully lit up when displaying the number 8.

Lesson 9: Let's explore Edison's musical talents

Estimated time	45-60 minutes
Additional resources needed	nil

Overview

Simple musical tunes are a great example of sequence in action. Using Edison to create music ties together sequence and inputs-outputs into programs that students love.

This activity is designed to mix the familiar with the unknown. By having students experiment using new blocks without explicit instructions or detailed explanations, they apply computational thinking practices and develop problem-solving mindsets.

Tips and tricks

- The song used in task one in the worksheet is *Mary Had a Little Lamb*. Use this share-code to access a program with more of the song: https://www.edscratchapp.com?share=Eb12x3Dm

- The third input parameter in the ‘music note’ block has three choices: (-), sharp, flat. The default (-) input means to play a normal note, not a sharp or flat.
- The ‘tempo’ block sets the tempo to be played. Only blocks which come after a ‘tempo’ block will be affected.
- Edison’s default tempo is ‘medium’. If no ‘set music tempo to’ block is used, the robot will play notes at the default tempo.
- Use this share-code to access a program with more of *The Hokey-Pokey* song: https://www.edscratchapp.com?share=v0wMx5D5
  NB: This version of the song uses loops.
  - You may want to wait to use this program until after students have studied repeating loops.
  - Repeat loops cannot be used inside ‘play music in background’ blocks. To use the longer version of the song in the background, duplicate the blocks inside the loop and add them into the program in sequential order instead.

Answer key

Question	Type	Answer/Sample answer	Marking notes
1	SE	*Very fast*	
2	SE	*It made the song play a lot quicker compared to the original with the ‘medium’ input parameter.*	
3	SE	*The ‘set tempo’ block sets the speed or tempo of the music.*	Students should identify that the block sets the tempo of the music.
4	EA	*If the ‘set tempo’ block is at the end, it won’t do anything. This is because Edison looks at the blocks one by one and does them in the order. If there are no music blocks after the tempo block, then the tempo block has no effect.*	Ideally, students will identify that sequential programming means things only go into effect in order. Consider having students try using the tempo blocks in different places in the program to test and reinforce this concept.
5	EA	*No, it played the music and then moved. I think this is because the blocks are in*	Student answers should note that the robot doesn’t

		order with all the music first and then the drive blocks.	move until after the music plays and identify sequence as the reason.

Lesson 9 - BONUS Challenge: You are the conductor

Estimated time	30-45 minutes
Additional resources needed	Sheet music or access for students to look up songs

Overview

This free-form activity asks students to bring their favourite tune to life by programming it in EdScratch for Edison to play. Music is a great way for kids to express themselves and their interests, and this activity lets them marry that enthusiasm with programming, helping showcase the arts in STEAM (science, technology, engineering, arts and maths).

Tips and tricks

- Having students search online for their favourite music opens up opportunities to discuss safe search habits, copyright and attribution, and many other digital citizenship topics.
- Students with musical talent can choose to write their own original song to program instead of using an existing song.

Lesson 10: Let's explore bugs and debugging

Estimated time	30-45 minutes
Additional resources needed	nil

Overview

Brace yourself – this lesson has a lot more text and a lot more definitions than most, but don't let that scare you away! Understanding bugs and debugging is one of the most critical skills you need your students to develop. Computational thinking is, in many ways, all about problem-solving. And problem-solving in computer science is all about debugging.

Students and teachers alike often struggle when moving from purely sequential programming to coding using more dynamic programming structures such as loops, branching and conditionals. A key reason people struggle is that they lack the debugging skill-set needed to find, identify and fix errors successfully.

The fundamentals of debugging are introduced in this lesson, and the practice is applied to the EdScratch environment. This mainly offline activity will begin to develop debugging capabilities in your students that are crucial for setting them up for success in activities in other units.

Tips and tricks

- While technology and computer science have developed the unfortunate reputation of being 'for males,' the history of computer science is full of fascinating people both male and female. Grace Murray Hopper (mentioned in this activity), Ada Lovelace, Margaret Hamilton, and Hedy Lamarr are just some of the fabulous females who have contributed to making modern computing what it is today. Any of these ladies make brilliant subjects for research projects and wonderful inspiration for students of any gender.
- You can find a complete guide to the warning messages in EdScratch at https://meetedison.com/robot-programming-software/edscratch/. This guide includes all warning messages, what they mean and examples of when you may encounter them. Warning messages in EdScratch can help you resolve any syntax errors and also give hints to help fix many common logical errors.
- All syntax errors that can be made in EdScratch will generate a red warning message in the bug box. Not all red messages are syntax errors, however.
- When a program runs but doesn't do what you expected, chances are there is a logical error in the program. Get into the habit of asking students what they want their program to do, then having them read out the program they have written in the same flow that the robot will. This will often uncover the logical error.

Answer key

Question	Type	Answer/Sample answer	Marking notes
1	EA	Syntax error: *the tempo block cannot go inside of the 'play music in background' grouping block. Only music note blocks can go in that block.* Logical error: *the problem is with the sequential logic the programmer used. The programmer needed to use the tempo block to set the tempo before the music blocks in order to have the music play at the very fast tempo.*	Any description that identifies and explains the two errors correctly is acceptable.
2	EA	*To get the program to work as described the following changes need to be made:*	Student answers should identify and fix all five

		1 – the distance input in the 'spin right' block needs to be 360, not 630. [logical error] *2 – the speed inputs for the 'spin left' and 'spin right' blocks need to be the same value. [logical error]* *3 – the 'set tempo' block needs to be outside of the 'play music in background' grouping block [syntax error]* *4 – the 'set tempo block' needs to be before the 'play music in background' block [logical error]* *5 – the 'spin left' and 'spin right' blocks need to be after the 'play music in background' block [logical error]*	errors. The inclusion of the error type is not requested in the worksheet but may be helpful in reviewing the bugs with students.

Lesson 11: Let's explore Edison's motors

Estimated time	45-60 minutes
Additional resources needed	nil

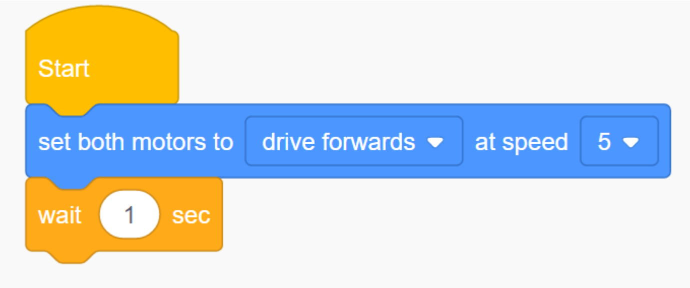
Overview

This lesson pulls together many of the key concepts from earlier lessons, including sequential programming, inputs and outputs, input parameters and debugging. Students further explore how Edison's motor outputs are controlled through drive blocks and learn that the motors can operate, and be controlled, independently. They are given a creative thinking challenge to imagine a use for this independent motor movement in a robotics application.

This activity also has students apply their understanding of how inputs (including information input through input parameters) affect outputs. They use this understanding to explore why some blocks in the language are dependent on other blocks. They are then challenged to apply their knowledge to get a program to work in an activity designed to push them, just a little bit, out of their comfort zone.

Tips and tricks

- The 'set motors' blocks only turn the motors on. These blocks require additional blocks to control the motor's activity by establishing a condition for the block. For example, this program says to turn the motors on, set to move forward, then says to wait in that condition for 1 second.



```
Start
set both motors to drive forwards at speed 5
wait 1 sec
```

- The ‘set motors’ blocks can be a bit tricky to use at first, but they are actually very powerful blocks and are very helpful in many different robotics build projects using Edison.
- Even if you do not plan on using the ‘set motor’ blocks in other activities in this unit, this lesson activity is a good opportunity for students to experience, and overcome, difficulties in programming.

Answer key

Question	**Type**	**Answer/Sample answer**	**Marking notes**
1	EA	*The left wheel moves backwards.*	
2	EA	*The right wheel moves forwards.*	
3	SE	*I think that the ‘set left motor’ and the ‘set right motor’ blocks are the blocks that control only one motor because they say which motor is being affected.*	Students may not select the correct blocks, but they should have a reason to support the choice they make. Students may have discovered that the input parameters for the ‘stop’ block can also be set to affect only one motor at a time.
4	SE	*I would turn Edison into a fan, and the one motor would spin the fan blades.*	This is an exercise in

<table>
<tr><td></td><td></td><td></td><td>the application of creative thinking. Any idea that uses one of Edison’s motors is acceptable .</td></tr>
<tr><td>5</td><td>EA</td><td><table>
<tr><th><strong>Information</strong></th><th><strong>In the program?</strong></th><th><strong>Value</strong></th></tr>
<tr><td>Direction</td><td><em>yes</em></td><td><em>forwards</em></td></tr>
<tr><td>Distance</td><td><em>yes</em></td><td><em>3</em></td></tr>
<tr><td>Distance units</td><td><em>yes</em></td><td><em>cm</em></td></tr>
<tr><td>Speed</td><td><em>yes</em></td><td><em>4</em></td></tr>
</table></td><td></td></tr>
<tr><td>6</td><td>EA</td><td><em>The robot drove forwards for 3cm at the speed of 4.</em></td><td></td></tr>
<tr><td>7</td><td>EA</td><td><table>
<tr><th><strong>Information</strong></th><th><strong>In the program?</strong></th><th><strong>Value</strong></th></tr>
<tr><td>Direction</td><td><em>yes</em></td><td><em>forwards</em></td></tr>
<tr><td>Distance</td><td><em>no</em></td><td></td></tr>
<tr><td>Distance units</td><td><em>no</em></td><td></td></tr>
<tr><td>Speed</td><td><em>yes</em></td><td><em>5</em></td></tr>
</table></td><td></td></tr>
<tr><td>8</td><td>SE</td><td><em>The robot didn’t do anything. It couldn’t because it did not have all the information it needed.</em></td><td>The robot will not have the information to know how far to move, so will appear to do nothing. Student answers may note that there is a</td></tr>
</table>

			message in the bug box warning them that they need additional blocks.
9	SE	*First, I tried adding a 'set music tempo' block to the start of the program, but it still didn't move. I moved the 'tempo' block to the end of the program but that still didn't work. I looked at the other Drive blocks and saw that seconds was one of the distance units. That made me think of the 'wait' block, so I added that to the end of the program and it worked!* *I was getting really frustrated when I couldn't get the program to work, but once I did figure it out I was really happy!*	The key here is developing resilience. Any path students took and whether or not they got the program to work is secondary.

Lesson 11 - BONUS Challenge: Spinning garden

Estimated time	N/A - Project
Additional resources needed	Maker-space/crafting supplies, EdCreate kits/LEGO bricks

Overview

This open-ended project has students apply what they have learned about sequential programming, inputs and outputs, and debugging into a physical computing project. This robotics build project asks students to create a spinning garden. Students need to plan their garden, build the different elements to attach to the robots and program each robot to spin its attachment.

Tips and tricks

- You can see a version of a spinning garden project using Raspberry Pi boards on YouTube at https://youtu.be/4Fs7y7gZIag?t=424 (Jump to the 7:00-minute mark to see the final spinning build in action.)
- Part of the idea of this project is to turn the robots onto their sides and use the robot's motors in a way other than to 'drive'. However, this project is also about unleashing students' creative thinking and problem-solving. If students decide to use the robots in ways other than with the robots on their sides, but are still using them to create a spinning garden, that is fine!

Lesson 11 - BONUS Challenge: Spinning solar system

Estimated time	N/A - Project
Additional resources needed	Maker-space/crafting supplies, EdCreate kits/LEGO bricks

Overview

This project has students apply what they have learned about sequential programming, inputs and outputs and debugging into a physical computing project. This robotics build project asks students to create a model solar system. Students need to plan their approach, build the different elements to attach to the robots and program each robot to spin its attachment.

Tips and tricks

- If you want your solar system to be a more accurate model, work with students to create the different scales needed to develop a more representative model. For example, determine what scale the planets are sized to and the scale for the distance each planet needs to be spaced from one another and the sun.
- If you want to allow for more creativity, have students design their own star system instead of making a model of the solar system. Students can create and name their own planets, and decide on the composition of each planet including features like its dimensions, colouring, habitability, etc. Have students write stories about this star system and any inhabitants which reside inside of it.

Skill challenge option 1: Cartographer and navigator

Estimated time	N/A - Project
Additional resources needed	Maker-space/crafting supplies for making the map

Overview

Get your students to apply sequence to a project where they design the set-up, the challenges, and the solutions. Students create a map large enough to use with their Edison robots, then design programming challenges to be solved using that map.

Designing the programming challenges and testing them for viability gets students to work with sequence and the other concepts from this unit in a new way. This project can be used as a ‘showcase’ for students to demonstrate their learnings from the unit.

Tips and tricks

- Consider linking this project to any relevant geography or world studies the students are doing by having students make maps of places they are learning about in other subjects.
- If students make their map using a grid system, it will be easier to navigate, but might not be as accurate. This is a good example of trade-offs in design and makes for a good example to discuss the topic.
- Working with multiple students? Try this variation: have the older students create a grid-system map and different driving challenges. (You may find it best to keep it simple and skip using 'Sound' and 'LED' elements for these challenges.) The 'cartographers' can then work with the younger students in the role of mentors, helping the other students to solve the challenges. The younger students can either use a push-button robot, program Edison using the EdBlocks programming language or help the older students program the robot using EdScratch. Whichever way you do it, giving your students the chance to help teach the concept of sequence will cement this key computation thinking concept for them.

Skill challenge option 2: Writer and director

Estimated time	N/A - Project
Additional resources needed	Maker-space/crafting supplies for making the story map

Overview

Story maps are helpful tools both for writing stories and for breaking down the stories we read to show comprehension. Story maps are great examples of sequence in the real-world and that connection is the basis for this open-ended capstone project.

Students create a story using a story map. They 'illustrate' the story using Edison – by programming the robot to help bring the story to life! Students should use a combination of Edison's outputs using the robot's motors, LEDs and sounds to highlight key moments in their story, add lighting or sound effects or however else they like.

Tips and tricks

- A 'story mountain' style story map works well as a base design for this project as it creates a nice linear pathway. You can see an example of this style of layout here https://www.tes.com/lessons/wImfQ5vQh-8OPA/sequencing
- This project lends itself well to mixed media by filming presentations – a great 'showcase' opportunity.

What next?

Completed these lessons and keen to continue your coding journey? You can keep learning with EdScratch and Edison, learning new coding skills as you go.

You can find all the EdScratch lessons at https://meetedison.com/robot-programming-software/edscratch/#EdScratch-resources

If you've finished all of this unit, try the next unit *Got loops?* Available at https://meetedison.com/content/EdScratch/EdScratch-student-lesson-activities-Unit-3.pdf