Debugging 1

Ages: 5-7
Length: 1 hour
Equipment: PDF print out

Introduction

This lesson introduces the relevant words; **_debugging_** through activities that link to student's lives to help them relate to the concepts introduced. This lesson will build upon prior learning relating to building precise algorithms.

Curriculum Alignment	❤ UK National Curriculum Computing Key Stage 1
Learning Objective	❤ To identify an error ❤ To demonstrate how to find and fix an error in code
Keywords	❤ Debug ❤ Error
Resources	❤ Debugging cards worksheet and solution ❤ Anagrams worksheet and solution
Lesson Sections	❤ Setting the Scene ❤ Activity 1 – identifying an error in a sequence ❤ Activity 2 – MiRo simulator ❤ Summary

Setting the Scene

How do we find an error?

We need to look at the sequence to find what is not right so it can be fixed.

Concept	How it is used
We need to look at the problem to find the **_error_** so it can be fixed. *Can you think of a problem you faced where you needed to find the error?*	When you write code, you are building a sequence. If there is an error in the sequence it needs to be found and fixed.

Pattern recognition is a skill that can help locate an error, for example an error in a sequence.

First let us look at how to identify an error.

Activity 1

Activity 1 is focused on getting students to think about what an **_error_** is, how to **_debug_** it and relate this to their everyday tasks.

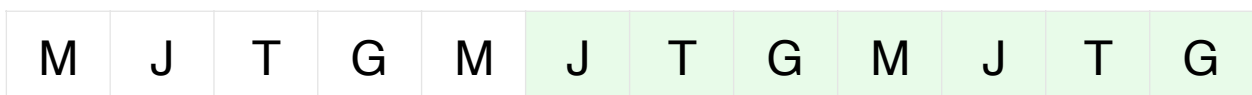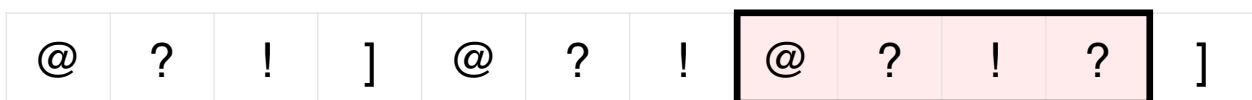| Scenario 1	Scenario 2
You have been given a maths task and you and your friend have two different answers. How do you find who is right? 1. Complete math task again 2. Check numbers added in are the same 3. Check the correct math operator has been used	You are playing a new game on a games console and you think A is to jump but it is not working. What would you do to find the error? 1. Read the instructions 2. Press A and see what happens 3. Press other buttons in turn to find out which jumps

> Where else do you debug a problem?

Whole Class Activity

Explain to the group that half of them will watch a dance routine and recreate it. They will only be able to watch it once and then attempt to recreate it. The other half will be able to watch the clip again and offer feedback on how to fix the dance routine to get it the same as the clip. The first 30 seconds of this clip (**https://www.youtube.com/watch?v=JWTyO8npkOQ**) would be a good starting point.

Small/Individual Group Activity

Split the students into small groups or pairs and give them the *debugging pattern cards*.

@	?	!	]	@	?	!	@	?	!	?	]

@	?	!	]	@	?	!	]	@	?	!	]

M	J	T	G	M	T	G	M	J	T	G	M

M	J	T	G	M	J	T	G	M	J	T	G

In your group, I want you to look at the patterns in front of you. *Without any information can you work out what the **error** is?*

Encourage discussion within the group, ask them if they found it easy to find the **error**? What made it easy to find the error? Can they fix it?

You use ***debugging*** daily without thinking when you are faced with a new challenge. You may not get it right first time and need to learn from the errors you make.

Differentiation

If pupils need extra support to understand what debugging is, show the results for one set of cards in the right order to help visualise the end solution.

For higher ability pupils ask them to support other pupils as 'debugging' assistants by giving hints not full solutions.

Activity 2

This activity is based on MiRoCODE. Guide students to their computers and to start a new workspace either individually or in pairs. It instructs students in how to create and run programs using procedures.

Any blocks used in your program could need ***debugging*** if an error occurs. Depending on the program you create, depends on the blocks you use.

When you create a program you should plan it, create it and then test it. When you test a program you are testing if it works as you want it to. If it does not work then you start to look at the sections of code to find the error and fix it.

Why should you test a program as you develop it?

Example answer:
So you can see if it works the way you want it to and to find any errors. You can then debug these errors to help your code work properly.

I have some code that is not working, can you help and debug it?
We are going to recreate the code and debug and fix it.

There are two pieces of code that allow MiRo to simulate 'YES' by nodding its head and 'NO' by shaking its head; but something is wrong, can you help?

For advanced students, the Step-by-Step guide can be left off and they can work their way through the steps on their own.
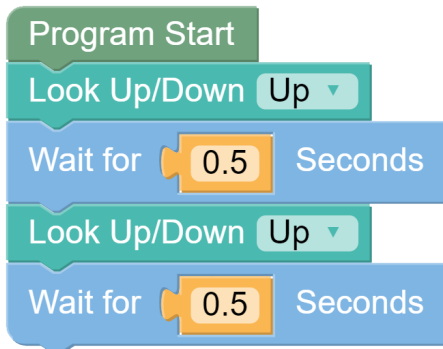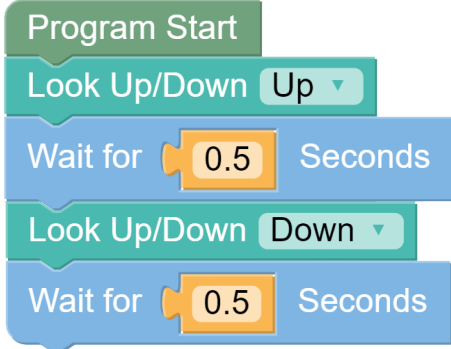
Algorithm planning	
YES	**NO**
A. Lift head UP B. Lift head DOWN	A. Look LEFT B. Look RIGHT

What does the code look like? Can you see anything wrong as you create it?

Using this algorithm as your plan, can you re-create the program in the MiRoSIM.
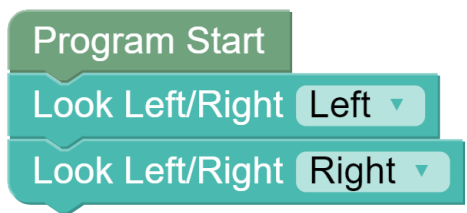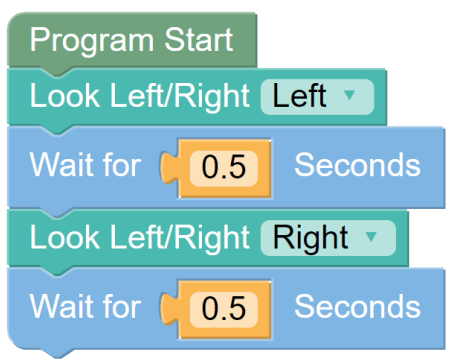

Activity 2 - Program solution YES

NB. There can be more than one solution in programming, encourage experimentation and use solution below as a guide

Zoom in to see MiRo's face.	** This is optional for higher ability pupils as this may be tricky for younger users.
Step 1-6 ❥ Add 'program start' from 'Setup' ❥ Add 2 x 'Look up/Down [up]' blocks from 'Simple Motion' ❥ Add 2 x 'wait for 1 seconds' block from 'Control' ❥ Set both wait blocks to 0.5 seconds	** The error is that both blocks are set to 'up' so will not go up and down. Encourage students to try and debug the code and test it to find the error. Program Start Look Up/Down Up ▾ Wait for 0.5 Seconds Look Up/Down Up ▾ Wait for 0.5 Seconds
Step 7 ❥ Click Simulator Play OR ❥ Click Robot play	** If you are using the physical MiRo ensure the IP address is correctly added to the onscreen code to allow the code to be communicated to MiRo to action. ** If you are using the simulator it is recommended to zoom in to see MiRo clearly but not essential
Solution	Program Start Look Up/Down Up ▾ Wait for 0.5 Seconds Look Up/Down Down ▾ Wait for 0.5 Seconds

Activity 2 - Program solution NO

NB. There can be more than one solution in programming, encourage experimentation and use solution below as a guide

Zoom in to see MiRo's face.	** This is optional for higher ability pupils as this may be tricky for younger users.
Step 1-6 🐰 Add 'program start' from 'Setup' 🐰 Add 2 x 'Look Left/Right [Left]' blocks from 'Simple Motion' 🐰 Set to be 'left' then 'right'	** The error is that both blocks do not have the wait block in between the blocks to allow the code to run. Encourage students to try and debug the code and test it to find the error. Program Start Look Left/Right Left ▾ Look Left/Right Right ▾
Step 7 🐰 Click Simulator Play OR 🐰 Click Robot play	** If you are using the physical MiRo ensure the IP address is correctly added to the onscreen code to allow the code to be communicated to MiRo to action. ** If you are using the simulator it is recommended to zoom in to see MiRo clearly but not essential
Solution	Program Start Look Left/Right Left ▾ Wait for 0.5 Seconds Look Left/Right Right ▾ Wait for 0.5 Seconds

Summary

Have a discussion with the class about what they have learnt in the lesson. Discuss the new word learnt: ***debugging*** and talk through any difficulties they had.

Ask students to complete the self-assessment and can be done by thumbs up, down and centre or using the images; on the following 3 questions

Questions
Can you describe what debugging is?
Can you give an example of how debugging is used?
Can you debug a given program in MiRo simulator and fix it?