

3-C

# TEACHER'S HANDBOOK

Lesson Plans, Guides & Support



GETTING LOOPY  
Using Python



# CONTENTS PAGE

Section	Page Number
Breakdown of Lesson Pack	4
Developing Digital Leaders	5
How to Use this Handbook	6
<b>Lesson 3.9 A Loopy Introduction</b>	<b>7</b>
Lesson Plan	7
Links to the Curriculum	9
Teacher Guides	12
PowerPoint Notes	13
Solutions	16
<b>Lesson 3.10 Remotely Operated Marty</b>	<b>19</b>
Lesson Plan	19
Links to the Curriculum	21
Teacher Guides	25
PowerPoint Notes	26
Solutions	29
<b>Lesson 3.11 Guess the Number</b>	<b>31</b>
Lesson Plan	31
Links to the Curriculum	32
Teacher Guides	36
PowerPoint Notes	37
Solutions	39
<b>Lesson 3.12 Guess the Number Continued</b>	<b>41</b>
Lesson Plan	41
Links to the Curriculum	42
Teacher Guides	46
PowerPoint Notes	47
Solutions	48

# BREAKDOWN OF LESSON PACK

Learners will continue to explore different coding concepts that they may have already seen in Scratch but using Python with Marty the Robot. Students will be introduced to loops in Python where they will be able to explain the difference between for and while loops with an understanding of when to use each based on the context of the program.

## Core Concepts

- ✓ Sequences
- ✓ Loops
- ✓ Debugging
- ✓ Algorithms
- ✓ Application of Engineering
- ✓ Variables
- ✓ Conditionals

## Core Skills

- ✓ Collaboration
- ✓ Computational Thinking
- ✓ Problem Solving
- ✓ Communication
- ✓ Digital Literacy
- ✓ Creative Thinking
- ✓ Critical Thinking & Analysis

### 3.9 A LOOPY INTRODUCTION

Students begin to explore repetition in their programs and consider the different methods they might use to reduce this. They will need to critically analyse and differentiate shapes in order to create a shape guessing program.

### 3.10 REMOTELY OPERATED MARTY

Investigating remotely operated robots and vehicles, students will understand the use cases for this kind of technology. Students can then design their own remotely operated robot by programming a text-based controller.

### 3.11 GUESS THE NUMBER

Continuing to explore alternative approaches to writing loops in their programs, students will use conditionals to create a game called guess the number. Students will need to use Marty's physical features to give feedback to players.

### 3.12 GUESS THE NUMBER CONTINUED

Continuation of Lesson 3.11 where students will begin to analyse their games to design further challenges and customise their guess the number program.

# DEVELOPING DIGITAL LEADERS

Our students are living in a digital world. We introduce core concepts throughout our lessons such as algorithms and if statements but we want to emphasise the importance of developing digital leaders who are engaged and can thrive in a connected, digital world. This is why in each of our lesson packs, we will refer to the *ISTE Standards for Students\** to highlight how our lessons help guide the way.

Empowered Learner	
1-a Set Learning Goals	✓
1-b Customise Learning	✓
1-c Seek Feedback	✓
1-d Explore Technology	✓

Digital Citizen	
2-a Digital Identity	
2-b Online Behaviours	✓
2-c Respect & Sharing	
2-d Personal Data	

Knowledge Constructor	
3-a Effective Research	✓
3-b Evaluate Resources	✓
3-c Curate Information	✓
3-d Explore Real Issues	✓

Innovative Designer	
4-a Use a Design Process	✓
4-b Select Digital Tools	
4-c Prototype Iteration	✓
4-d Problem Solving	✓

Computational Thinker	
5-a Formulate Problems	✓
5-b Data Analysis	
5-c Decomposition	✓
5-d Algorithms	✓

Creative Communicator	
6-a Tool Selection	
6-b Original vs Remix	✓
6-c Communicate Ideas	✓
6-d Presentation	✓

Global Collaborator	
7-a Diverse Teams	✓
7-b Collaborative Tech	✓
7-c Giving Feedback	✓
7-d Investigate issues	✓

# HOW TO USE THIS HANDBOOK

We have created this handbook so that you have everything you need to deliver our lesson packs. For each lesson you will have the following information,



## LESSON PLANS

Outcomes, Resources & Learning Plans



## LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks



## TEACHER GUIDES

Prompts & Questions for Delivery of Lessons



## POWERPOINT NOTES

Slide Notes to Deliver our PowerPoints



## SOLUTIONS

Sample Solutions & Activity Guides

# A LOOPY INTRODUCTION

Outcomes, Resources & Learning Plans

**EDUCATION LEVEL:** Third/Fourth Level (Ages 11-15)

**PRE-REQUISITES:** 3.1, 3.2, 3.5, 3.6

**LESSON DURATION:** 45 minutes

**DEVICE COMPATIBILITY:** Laptop or PC

**CROSS-CURRICULAR LINKS:** Technologies/Literacy/Numeracy

## LESSON OVERVIEW

As students have been exploring Python with Marty, they may have noticed a lot of repetition in their lines of code. As a class, students will explore where this repetition might happen and try out different methods to reduce the number of lines in their programs. Students will create a shape guessing program for Marty.

### LEARNING OBJECTIVES

- Highlight where a loop/repeat statement could be used to reduce repetition
- Examine the flow of their programs and draw flowchart diagrams
- Write a for loop in Python with the correct parameters for the purpose

### KEY VOCABULARY

- Python
- Repeat / Loop
- For Loop
- Repetition
- Flowchart

### RESOURCES & EQUIPMENT

- Marty the Robot
- Student workbooks (Lesson 1)
- Devices with a Python editor and *martypy* installed

### ADDITIONAL READING

- Marty the Robot Educator Guide
- Educator FAQ
- Getting Started with Python
- MartyPy Documentation

## LEARNING PLAN & ACTIVITIES

1. Class discussion of the ways that we can reduce the amount of repetition in our programs
2. Challenge students to write instructions for coding someone to walk in a square and highlight where the repetition appears
3. Introduction to loops as a way to repeat the instructions and highlight structure of a flowchart for a loop
4. Show students example instructions with the challenge to pair the instruction up to the correct shape
5. Script example on how you can use loops in Python with special attention given to the indentation required for loops (similar to if statements)
6. Challenge students to create a shape guessing program that asks the user question, guesses the shape and shows the answer by coding Marty to walk out in that shape
  - A. Demonstration of script provided for this lesson to show students an example of tackling the problem
7. Demonstrate flowcharts for planning programs and encourage students to plan their programs first by drawing a flowchart and then starting to code in their small groups with Marty

## EXTENSIONS & CHALLENGES

- Extend the program to use the Python *Turtle* library to draw Marty's path on the screen (*Technologies/Literacy/Numeracy/Arts*)
- What is the least number of questions students could ask to narrow down the selected shape? (*Technologies/Literacy/Numeracy*)

# LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

## Curriculum for Excellence - Technologies

● = Fully Addresses Benchmark

○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.9
Digital Literacy	TCH 0-01a	●
	TCH 1-01a	●
Technological Developments in Society and Business	TCH 0-05a	●
	TCH 1-05a	○
Craft, Design, Engineering and Graphics	TCH 0-11a	○
	TCH 0-12a	●
	TCH 1-12a	●
	TCH 3-12a	○
Computing Science	TCH 0-13a	●
	TCH 1-13a	●
	TCH 2-13a	○
	TCH 3-13a	●
	TCH 4-13a	○
	TCH 3-13b	○
	TCH 4-13b	●
	TCH 0-14a	●
	TCH 1-14a	●
	TCH 2-14b	○
	TCH 0-14b	○
	TCH 1-14b	○
	TCH 0-15a	●
	TCH 1-15a	●
	TCH 2-15a	●
	TCH 3-15a	○



# LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

## National Curriculum - Computing, Design & Technology

● = Fully Addresses Benchmark

○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.9
Computing	1-a	●
	1-b	●
	1-c	●
	1-e	●
	2-a	●
	2-b	●
	2-c	●
	2-f	○
	3-a	●
	3-b	○
	3-c	○
	4-a	○
	4-b	○
Design & Technology	1.1-a	●
	1.3-b	●
	2.3-b	○
	3.1-b	●

## Australian F-10 Curriculum - Digital Technologies, Design & Technologies

● = Fully Addresses Benchmark

○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.9
Digital Technologies	ACTDIK001	●
	ACTDIK002	●
	ACTDIP004	●
	ACTDIK008	●

# LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

## Australian F-10 Curriculum Continued...

● = Fully Addresses Benchmark

○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.9
Digital Technologies	ACTDIP010	●
	ACTDIP016	●
	ACTDIP019	●
	ACTDIP020	●
	ACTDIP029	●
	ACTDIP030	●
Design & Technologies	ACTDEP008	○
	ACTDEP009	●
	ACTDEP015	●
	ACTDEP025	○
	ACTDEP018	●

## CSTA K-12 - Computer Science

● = Fully Addresses Benchmark

○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.9
Computing Systems	1A-CS-01	●
	1A-CS-02	●
	1B-CS-01	○
	1B-CS-02	●
	2-CS-02	○
Algorithms & Programming	1A-AP-08	●
	1A-AP-09	○
	1A-AP-10	●
	1A-AP-11	●

# LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

## CSTA K-12 - Computer Science Continued...

● = Fully Addresses Benchmark

○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.9
Algorithms & Programming	1A-AP-12	●
	1A-AP-14	●
	1A-AP-15	○
	1B-AP-09	●
	1B-AP-10	●
	1B-AP-11	●
	1B-AP-13	●
	1B-AP-15	●
	2-AP-10	●
	2-AP-11	○
	2-AP-13	○
	2-AP-15	●

# TEACHER GUIDES

Prompts & Questions for Delivery of Lessons

## A LOOPY INTRODUCTION

### SPOTTING REPETITION IN YOUR CODE

Whilst looking through past Python projects, students might spot some repeating sections of code. Normally when we are writing out instructions in real life we might describe the action in detail the first time and simply say repeat this x number of times afterwards. If students have used Scratch previously then they might have already used loops to solve this problem of lots of repeating lines of code.

- What could we use to reduce the amount of repeating lines of code?
- What have we used in Scratch to do this?
- Is there a better way to describe these instructions using less lines?

### INTRODUCTION TO LOOPS

Some students will remember using loops whilst coding in Scratch and it is the same principle here when we are coding with Python. The only difference is the way that we say how to do a loop. Python is very precise about line indentation as it uses this to translate the code into another language the computer will understand.

- How did we create a loop in Scratch? What did it look like?
- What are the main differences between loops in Scratch and loops in Python?
- What do we need to look out for when we are writing loops in Python?

### FLOWCHARTS

When we are writing complicated programs, it is usually a good idea to plan them out before we sit down to write them. Flowcharts are one way of planning how our program is going to work and highlight to us the order of events happening. Note that you can still have problems with the *syntax* (layout and wording) of programs even when you follow a correct flowchart.

- What are flowcharts used for?
- Why do you think we should plan our programs first?
- Can we still find errors in our programs even if we follow the correct flowchart?

# POWERPOINT NOTES

Slide Notes to Deliver our PowerPoints

## Getting Loopy

Using Python

© 2010 Robotix Ltd. All Rights Reserved.  
You may print copies of this document but please do not redistribute or alter.

## Lesson 1 A Loopy Introduction

By the end of this lesson you will be able to,

- Highlight where a loop/repeat statement could be used to reduce repetition
- Examine the flow of your programs and draw flowchart diagrams
- Write a for loop in Python with the correct parameters for the purpose

Introduction to lesson outcomes

```
import martypy
marty = martypy.Marty('socket://192.168.8.122')

marty.celebrate()
marty.walk(2)
marty.celebrate()
marty.walk(2)
```

What could we use to reduce the amount of repetition in our Python programs?

How many students have written code like this with repetition? What can we do to make our code simpler?

## Example

In your workbooks, write instructions in English for someone to walk in a square

Now highlight where the repetition is in your instructions

Ask students to write instructions for someone to walk in a square in their workbooks. Encourage them to test it out and highlight where the repetition occurs.

## Introduction to Loops

In English

REPEAT 4:  
Repeat these actions 4 times { Wiggle  
Take 2 steps forward

Re-write your instructions to make use of a loop

Introduction to loops where we tell a group of movements/actions to repeat a specific number of times.

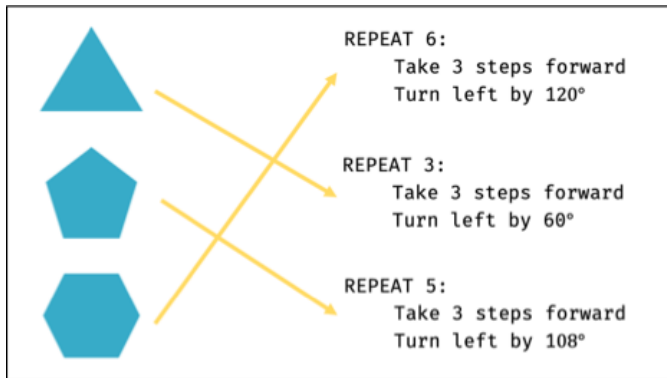


Match the Shape to the Correct Instructions

Students are tasked with deciding as a class which instruction matches to the correct shape

# POWERPOINT NOTES

Slide Notes to Deliver our PowerPoints



Reveals incrementally (animations in slides)

### How does it look in Python?

```
for x in range(4):
    marty.walk(2)
    marty.walk(7, 'auto', 120, 0, 1500)
```

What does this second walk command do?

Like if statements, we use indentation. Note that the second walk block with 0 step lengths allows turning on the spot.

### Write out your own definition of a loop & consider when you might need to use one

Write your own definition of a loop in workbooks and think about when we might need to use a loop with Marty

### Live Demonstration

#### Shape Guessing Program

Run example script to show students what they will be working on today

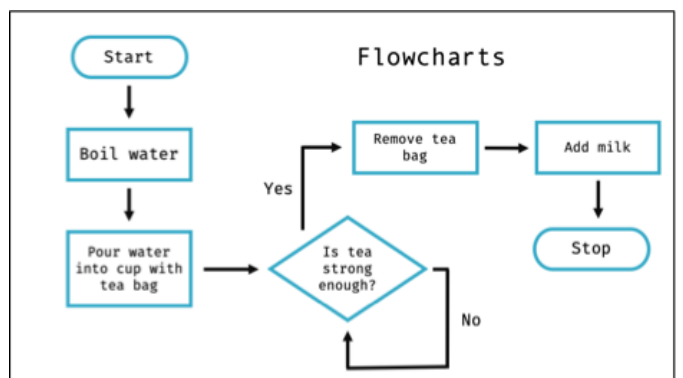
### But first, planning

Do you think we should plan our programs before we start coding?

How do you think we could do this?

What kind of information do you think we would need to plan?

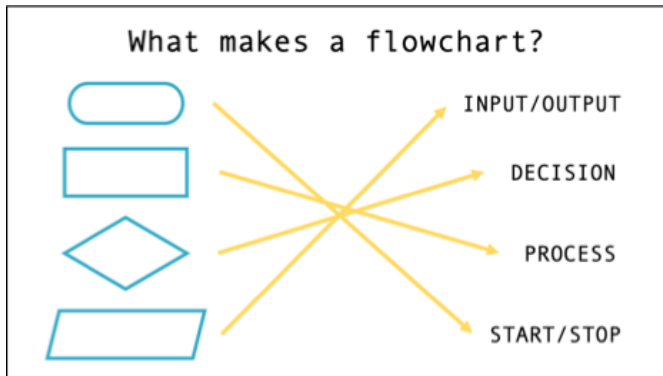
Should we plan our programs before coding and how?



What is this chart illustrating? What would you change about the diagram? Does everyone have milk in their tea?

# POWERPOINT NOTES

Slide Notes to Deliver our PowerPoints



Ask students to match the symbols to the function (animations in slide)

Write your own definition of a flowchart

Write a definition for a flowchart in the workbooks

Draw a flowchart for your shape guessing program

There is space in your workbooks

End of Lesson Reflection

One thing I enjoyed

What I found challenging

Complete reflections in student workbooks

# SOLUTIONS

Sample Solutions & Activity Guides

## 1 Instructions for walking in a square

In English, students should write out some basic instructions for walking in a square. Please note that they can't simply write walk in a square as we must assume that the person we are programming does not know what a square is. Therefore their instructions should be made up of commands such as *walk forwards 3 steps then turn 90 degrees to your right*

After writing these instructions out, students should spot repetition in their commands as they are repeating the same thing for each side of the shape. After discussing and introducing loops, students can re-write their instructions to include a loop. Here is an example of how this might look,

*REPEAT 4:*

*Walk forwards 4 steps*

*Turn left by 90 degrees*

## 2 Defining a loop

Students should use the space in their workbooks to write their own definition of a loop and consider examples when they might need to use a loop. Students will have a variety of slightly different answers but here is a sample definition.

*A loop is used to reduce the amount of repetition in your code so that you can repeat a specific set of actions or commands a number of times.*

Here are some examples of when you might use a loop:

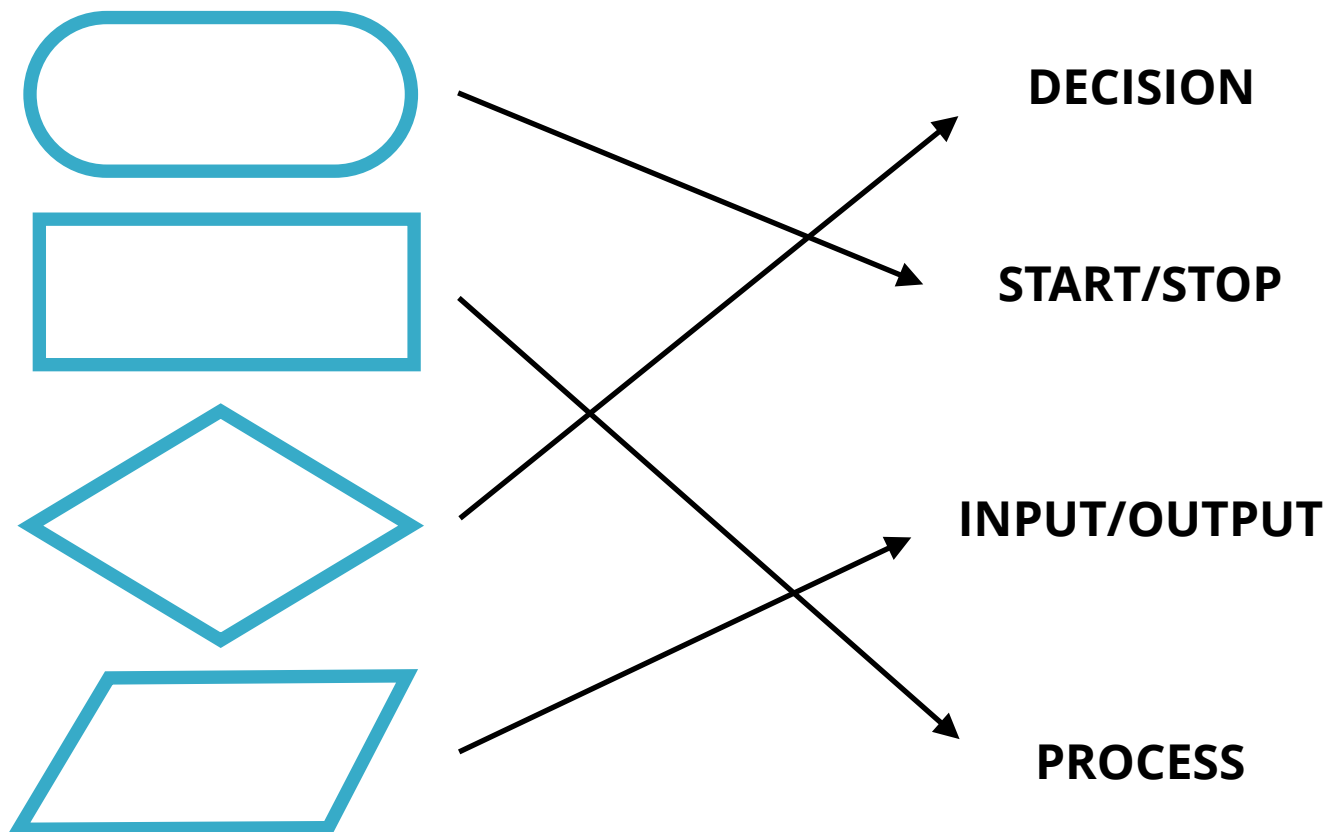
- Coding shapes or actions with repeating actions
- Asking the user to enter information several times such as when entering a password



# SOLUTIONS

Sample Solutions & Activity Guides

## 3 Breaking down flowcharts



## 4 Describing a flowchart

Here is a sample definition for a flowchart,

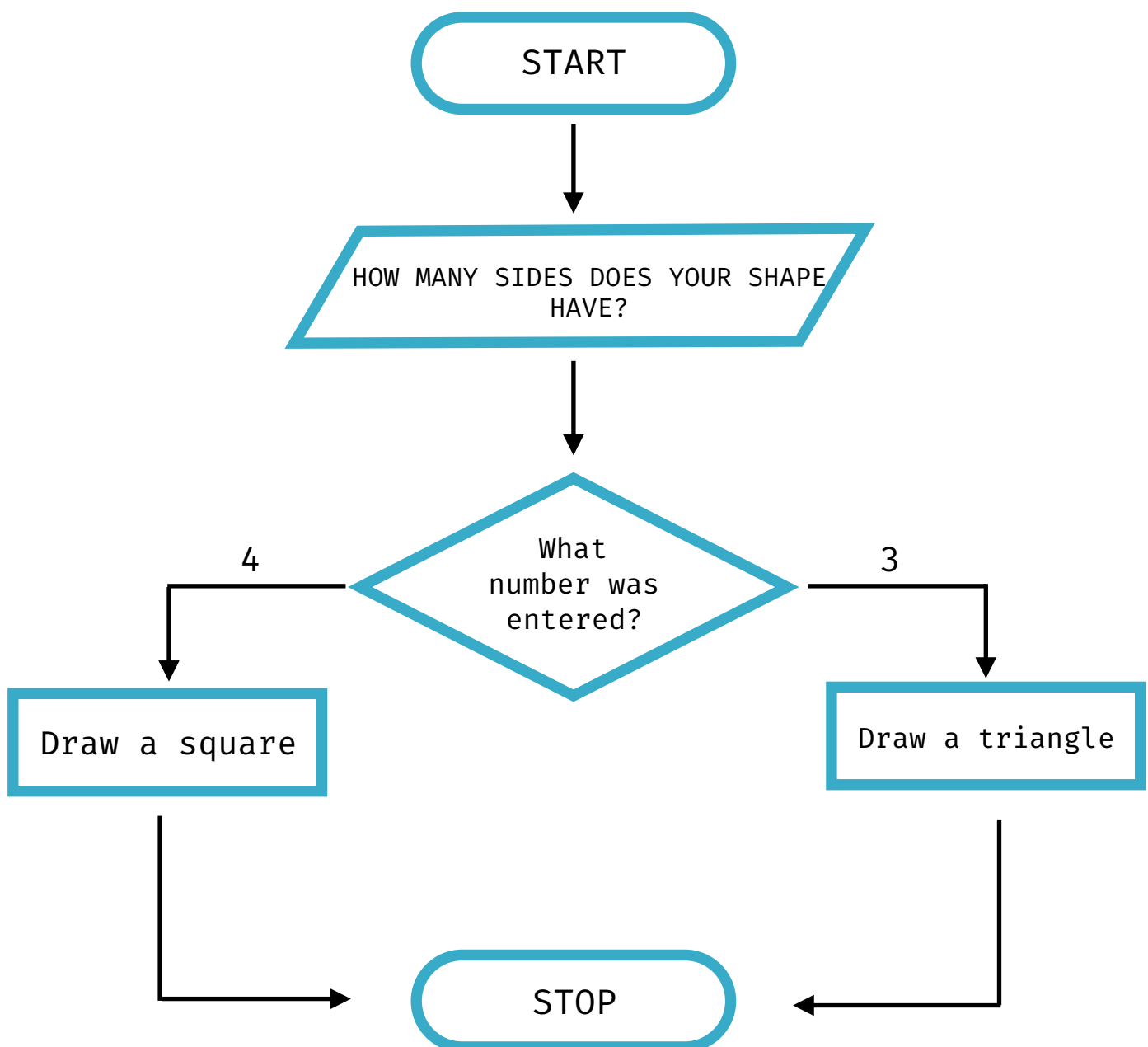
*A flowchart shows the sequence of movements/actions in an activity/program detailing when the system needs to ask the user for input, make a decision or complete a process. We can use flowcharts to help plan out our programs before we start coding to understand the sequence of the steps required to code our program.*

# SOLUTIONS

Sample Solutions & Activity Guides

## 5 Shape Guessing Program Flowchart

Please note that each student's flowchart may be slightly different depending on how they want to handle deciding which shape should be guessed. Here is an example,



# REMOTELY OPERATED MARTY

Outcomes, Resources & Learning Plans

**EDUCATION LEVEL:** Third/Fourth Level (Ages 11-15)

**LESSON DURATION:** 45 minutes

**CROSS-CURRICULAR LINKS:** Technologies/Literacy

**PRE-REQUISITES:** 3.1, 3.2, 3.5, 3.6, 3.9

**DEVICE COMPATIBILITY:** Laptop or PC

## LESSON OVERVIEW

We use robots and technology in many different places to explore the environment from a safe location. Students will be using a new type of loop, while loops, to create their own remotely operated Marty that can be controlled from a distance using text commands.

### LEARNING OBJECTIVES

- Explain the flow of a program that has a while loop
- Write a Python script using a while loop to repeat until a condition is met
- Draw a flowchart for a program with a while loop
- Explain why we might use robots for exploring new environments

### KEY VOCABULARY

- Loop / Repeat
- While Loop
- Repetition
- Condition
- Flowchart
- Remotely Operated Vehicle

### RESOURCES & EQUIPMENT

- Marty the Robot
- Student workbooks (Lesson 2)
- Devices with a Python editor and *martypy* installed

### ADDITIONAL READING

- Marty the Robot Educator Guide
- Educator FAQ
- Getting Started with Python
- MartyPy Documentation

## LEARNING PLAN & ACTIVITIES

1. Class discussion of what we might use a remotely operated robot or vehicle for (considering perhaps unsafe environments for humans to be in for long periods of time)
2. Challenge students to create a text-based remote control so that they can remotely operate their Marty from a distance to explore the classroom, running the demonstration file to show students what to expect
3. Highlight that they will need to use a loop that keeps running until the user types in the word *stop* - what programming concepts can they use to do this? How do they know how long the loop will need to run for?
4. Introduction to while loops as another type of loop that can be used in Python
  - A. Highlight the flowchart for this kind of loop along with how it looks in Python with focus on indentation and structuring a condition (similar to writing an if statement)
5. Students should plan out their programs using a flowchart and start programming in small groups with a Marty and device with a Python editor

# REMOTELY OPERATED MARTY

Outcomes, Resources & Learning Plans

**EDUCATION LEVEL:** Third/Fourth Level (Ages 11-15)

**LESSON DURATION:** 45 minutes

**CROSS-CURRICULAR LINKS:** Technologies/Literacy

**PRE-REQUISITES:** 3.1, 3.2, 3.5, 3.6, 3.9

**DEVICE COMPATIBILITY:** Laptop or PC

## EXTENSIONS & CHALLENGES

- Set up some webcams in another room with obstacles and challenge students to remote control Marty around the room whilst in a separate room using only the webcams (*Technologies/Literacy/Numeracy*)
- Research and present on technology that is used for exploring environments such as underwater or empty pieces of land (*Technologies/Literacy*)
- Research environments walking robots be useful for exploring? Would you design the robot to be like Marty? Ask students to present their findings on a poster (*Technologies/Literacy/Arts*)

# LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

## Curriculum for Excellence - Technologies

● = Fully Addresses Benchmark

○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.10
Digital Literacy	TCH 0-01a	●
	TCh 0-02a	●
Technological Developments in Society and Business	TCH 0-05a	●
	TCH 3-05a	●
	TCH 0-07a	○
	TCH 1-07a	○
Craft, Design, Engineering and Graphics	TCH 0-12a	○
	TCH 1-12a	●
Computing Science	TCH 0-13a	●
	TCH 1-13a	●
	TCH 2-13a	●
	TCH 3-13a	●
	TCH 3-13b	●
	TCH 0-14a	●
	TCH 1-14a	●
	TCH 2-14a	●
	TCH 0-14b	●
	TCH 1-14b	●
	TCH 0-15a	●
	TCH 1-15a	●
	TCH 2-15a	●
	TCH 3-15a	○
	TCH 4-15a	○

# LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

## National Curriculum - Computing, Design & Technologies

● = Fully Addresses Benchmark

○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.10
Computing	1-a	●
	1-b	●
	1-c	●
	1-e	●
	2-a	●
	2-b	●
	2-c	●
	3-a	●
	3-b	●
	3-c	○
	3-e	○
	4-a	●
	4-b	●
Design and Technology	1.1-a	●
	1.3-a	○
	1.3-b	●
	2.3-b	○
	2.3-c	○
	3.1-b	●

# LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

## Australian F-10 Curriculum - Digital Technologies, Design & Technologies

● = Fully Addresses Benchmark

○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.10
Digital Technologies	ACTDIK001	●
	ACTDIK002	●
	ACTDIP004	●
	ACTDIP010	●
	ACTDIP012	●
	ACTDIP013	●
	ACTDIP018	○
	ACTDIP019	●
	ACTDIP029	●
	ACTDIP030	●
Design & Technologies	ACTDEK001	●
	ACTDEP005	●
	ACTDEP008	●
	ACTDEP009	●
	ACTDEK010	○
	ACTDEP015	●
	ACTDEP025	●

# LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

## CSTA K-12 - Computer Science

● = Fully Addresses Benchmark

○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.10
Computing Systems	1A-CS-01	●
	1A-CS-02	●
	1B-CS-02	○
	2-CS-02	●
Algorithms & Programming	1A-AP-08	●
	1A-AP-09	●
	1A-AP-10	●
	1A-AP-11	●
	1A-AP-12	●
	1A-AP-14	●
	1A-AP-15	○
	1B-AP-09	●
	1B-AP-10	●
	1B-AP-11	●
	1B-AP-13	○
	1B-AP-15	●
	2-AP-10	●
	2-AP-11	○
	2-AP-12	○
Impacts of Computing	1A-IC-16	●
	1B-IC-18	●



# TEACHER GUIDES

Prompts & Questions for Delivery of Lessons

## REMOTELY OPERATED MARTY

### REMOTELY OPERATED VEHICLES/ROBOTS

Technology and robots are used all around the world to help us solve different problems or explore environments that might not be safe or easy for humans to navigate and explore. A simple Google search reveals the number of different robots in different roles from care to education and exploration.

For example, robots are used more and more for exploration of environments such as underwater where we can remotely operate the robot and use the camera to view what can be seen.

- What jobs/roles do you think robots play? Can you think of any examples?
- What kind of environments do you think we could use robots in for exploration? Why?
- What environments could we use Marty to explore remotely?

### TEXT-BASED REMOTE CONTROL

Students will be designing their own text-based remote control so that they can control what Marty is doing remotely. Students will need to consider what commands will be useful and incorporate this into their design.

- What commands will be useful to include in the remote control? Why?
- What programming concepts will you need to use?
- How are you going to handle commands being entered that are not supported?

### WHILE LOOPS

Students have already used a for loop in Python, however, for this project they will need to use while loops so that the loop keeps running until a condition is met.

- How would you describe a while loop?
- Why do you think we need to use a while loop and not a for loop like in Lesson 3.9?

# POWERPOINT NOTES

Slide Notes to Deliver our PowerPoints

## Lesson 2 Remotely Operated Marty

By the end of this lesson you will be able to,

- Explain the flow of a program that has a while loop
- Write a Python script using a while loop to repeat until a condition is met
- Draw a flowchart for a program with a while loop
- Explain why we might use robots for exploring new environments

Introduction to lesson outcomes

How might you describe what a remotely operated robot is?

Write down your description in your workbooks



What is a remotely operated robot? Consider examples

Why might we use robots to explore and do tasks instead of humans?

Write down some reasons in your workbooks



We have thought about tasks but why would we use robots instead of humans? Safety? Time? Efficiency?

Describe environments better suited for robots to explore?

Write down some keywords in your workbooks



Are there certain environments that are better for robots to explore rather than humans?

## Live Demonstration Text-Based Remote Control



Run the example script for the text-based remote control and discuss how the program works

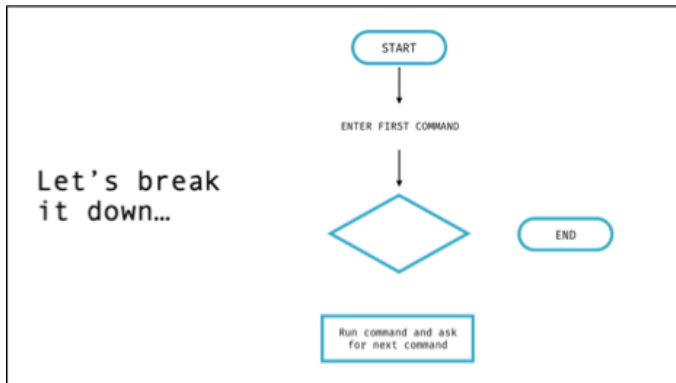
What coding concepts do you think we will need?

If Statements? Loops? Variables?

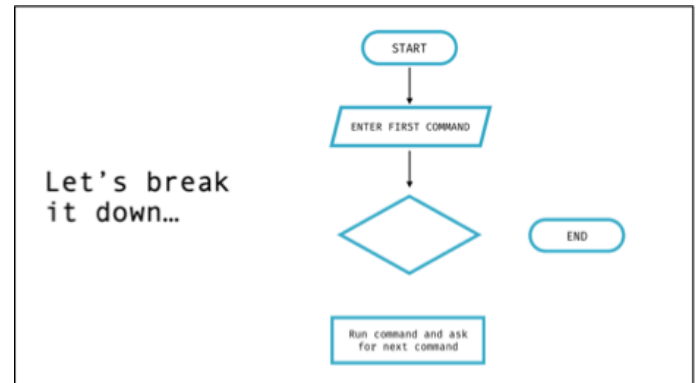
Discuss with students how they think the program works and what coding concepts are needed

# POWERPOINT NOTES

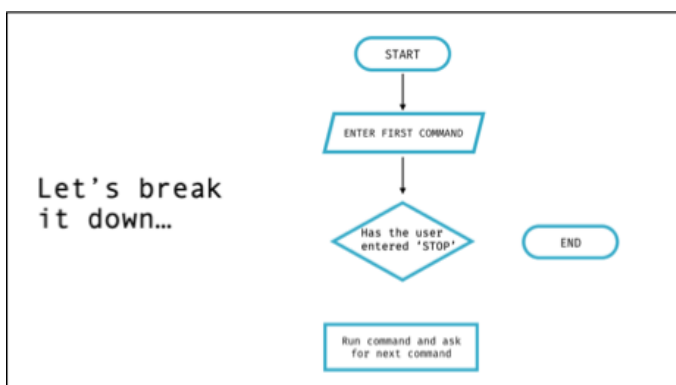
Slide Notes to Deliver our PowerPoints



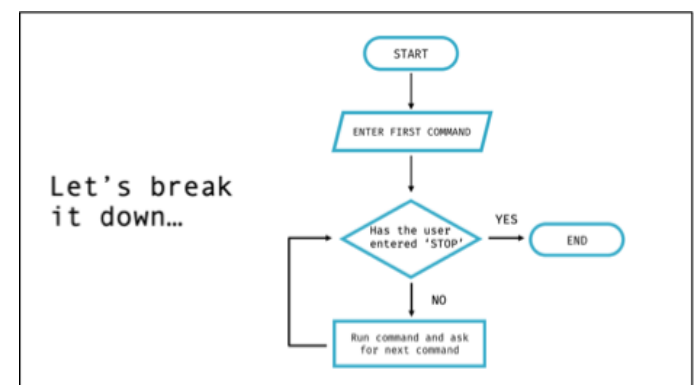
Lets use the flowchart to help us, step through the program and fill in the missing pieces of information



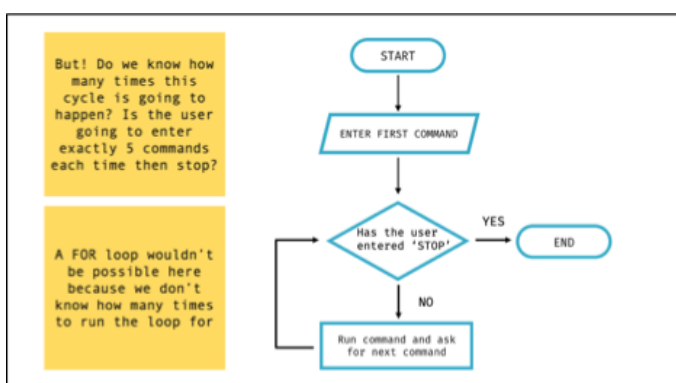
INPUT/OUTPUT block is needed to ask the user for a command to enter



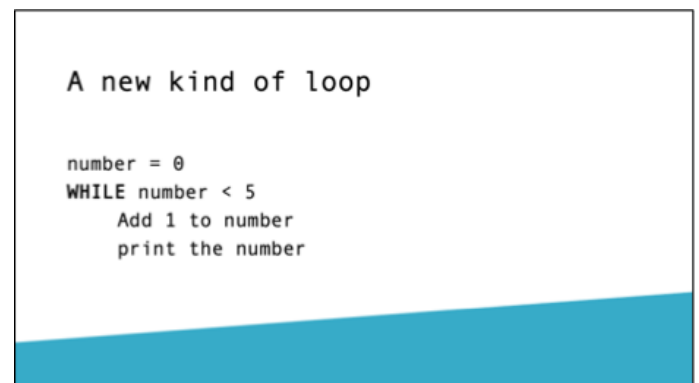
We check if the user has entered the STOP command



If the user has entered STOP then we END the program otherwise we run the command and ask for the next one



Do we know how many times the cycle will happen? We can't use a for loop here since we don't know that information



How would this loop work? When will it stop running?

# POWERPOINT NOTES

Slide Notes to Deliver our PowerPoints

## Back to our remote control

```
WHILE command doesn't equal 'STOP'  
  Run command on Marty  
  Ask for the next command
```

How would this loop work? When will it stop running?

## In Python

```
import martypy  
  
mymarty = martypy.Marty('socket://192.168.8.122')  
mymarty.hello()  
steps = 0  
  
while steps < 5:  
    marty.walk()  
    steps = steps + 1
```

What will happen if we run this program? Marty will keep walking until our variable has reached 5 (6 steps in total since we start at 0)

## Over to you...

Grab a device and a Marty and code your own text-based remote control using the flowchart we completed as a class earlier as a guide!

## End of Lesson Reflection

One thing I learned  
Something to ask next time  
One thing I want to research about remotely operated vehicles/robots

End of lesson reflection

# SOLUTIONS

Sample Solutions & Activity Guides

## 1 Discussing Remotely Operated Robots & Vehicles

Discussing as a class what remotely operated robots are and how we can use them to help solve different problems around the world, students should write down some of their ideas. Here are some examples to help out.

### **Reasons for using remotely operated vehicles/robots**

- Robots can explore unknown environments that we don't know are safe for humans
- Robots could complete certain tasks faster than humans
- We could use robots to explore certain environments for longer such as underwater

### **Environments remotely operated vehicles/robots might be used in**

- Underwater
- Space
- In the air

### **Definition of remotely operated vehicles/robots**

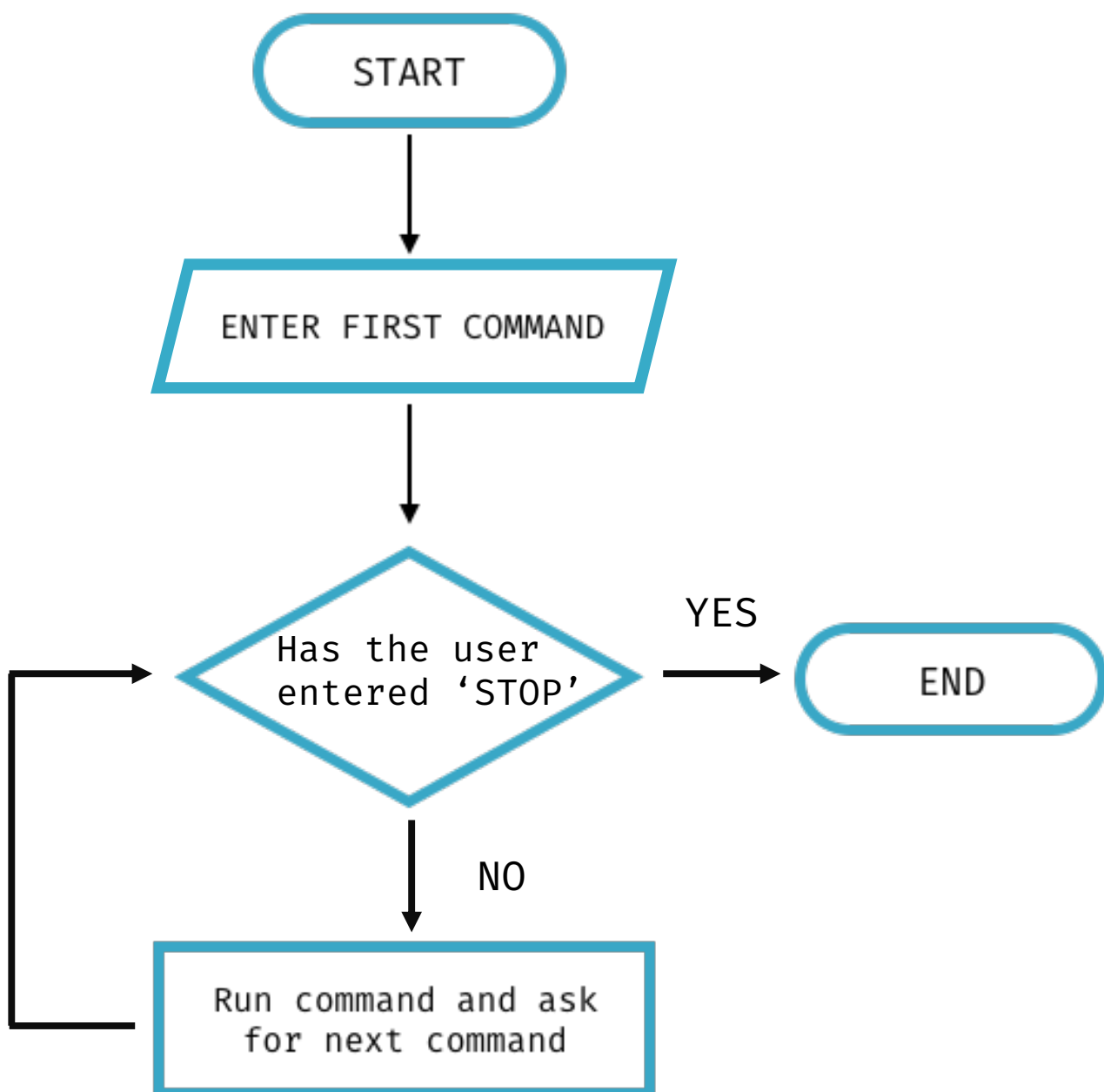
Remotely operated vehicles/robots allow for humans to control vehicles/robots to move around different environments to complete tasks or explore without humans having to physically be there. There is usually a camera attached to the vehicle/robot so we can see what is happening.

# SOLUTIONS

Sample Solutions & Activity Guides

## 2 Text-Based Remote Control Flowchart

This flowchart was highlighted and completed in the slides. Once students start to extend their programs they might have a different sequence of events.



# GUESS THE NUMBER

Outcomes, Resources & Learning Plans

**EDUCATION LEVEL:** Third/Fourth Level (Ages 11-15)

**PRE-REQUISITES:** 3.1, 3.2, 3.5, 3.6, 3.9, 3.10

**LESSON DURATION:** 45 minutes

**DEVICE COMPATIBILITY:** Laptop or PC

**CROSS-CURRICULAR LINKS:** Technologies/Literacy/Numeracy

## LESSON OVERVIEW

Students have now been introduced to two different types of loops in Python. Across the next two lessons students will design and build a number guessing game with Marty. To do this, students will need to differentiate between the two types of loops that have been covered so far.

### LEARNING OBJECTIVES

- Describe examples of when a for loop is needed
- Describe examples of when a while loop is needed
- Design and build a program that uses both a for and while loop
- Draw a flowchart for a program with nested loops

### KEY VOCABULARY

- For Loop
- While Loop
- Number
- Condition
- Control
- User Input

### RESOURCES & EQUIPMENT

- Marty the Robot
- Student workbooks (Lesson 3)
- Devices with a Python editor and *marty* installed

### ADDITIONAL READING

- Marty the Robot Educator Guide
- Educator FAQ
- Getting Started with Python
- MartyPy Documentation

## LEARNING PLAN & ACTIVITIES

1. Recap on what the students were working on last time and reflect on what they have learned about loops
  - A. What did you create in the last lesson? What coding concepts did you need to use?
  - B. How is a while loop different to the loop we used in Lesson 1 (3.9)?
2. Comparison of a while loop compared to the loop we used in the first lesson (called a for loop) in terms of how to write one in Python and the flowcharts for each
3. Given some example programs, students need to decide whether they would need to use a while or a for loop to solve the problem
4. Demonstrate to students the project they are going to be working on (example script provided) where they will need to design and build a guess the number game and discuss what coding concepts they think they will need to use
5. In small groups, students should design their program by drawing a flowchart before beginning to code with Marty and Python (note that students will continue this project in the next lesson so may not fully finish by the end of the lesson)

## EXTENSIONS & CHALLENGES

- See Lesson 3.12 for extensions

# LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

## Curriculum for Excellence - Technologies

● = Fully Addresses Benchmark

○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.11
Digital Literacy	TCH 0-01a	●
	TCH 1-01a	●
Technological Developments in Society and Business	TCH 0-05a	●
Craft, Design, Engineering and Graphics	TCH 0-12a	●
	TCH 1-12a	○
Computing Science	TCH 0-13a	●
	TCH 1-13a	●
	TCH 2-13a	●
	TCH 3-13a	●
	TCH 4-13a	○
	TCH 3-13b	●
	TCH 0-14a	●
	TCH 1-14a	●
	TCH 2-14a	●
	TCH 3-14a	○
	TCH 4-14a	○
	TCH 0-14b	●
	TCH 1-14b	●
	TCH 0-15a	●
	TCH 1-15a	●
	TCH 2-15a	●
	TCH 3-15a	●
	TCH 4-15a	○



# LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

## National Curriculum - Computing, Design & Technology

● = Fully Addresses Benchmark

○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.11
Computing	1-a	●
	1-b	●
	1-c	●
	1-e	●
	2-a	●
	2-b	●
	2-c	●
	3-a	●
	3-b	●
	3-c	○
	3-e	●
	4-a	●
	4-b	●
Design & Technology	1.1-a	●
	1.3-b	●
	2.3-b	●
	3.1-b	●

# LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

## Australian F-10 Curriculum - Digital Technologies, Design & Technologies

● = Fully Addresses Benchmark

○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.11
Digital Technologies	ACTDIK001	●
	ACTDIK002	●
	ACTDIP004	●
	ACTDIP010	●
	ACTDIP012	○
	ACTDIP013	●
	ACTDIP019	●
	ACTDIP028	○
	ACTDIP029	●
	ACTDIP030	●
Design & Technologies	ACTDEK001	●
	ACTDEP005	●
	ACTDEP008	●
	ACTDEP009	●
	ACTDEP018	●

## CSTA K-12 - Computer Science

● = Fully Addresses Benchmark

○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.11
Computing Systems	1A-CS-01	●
	1A-CS-02	●
	1B-CS-01	●
	1B-CS-02	●
	2-CS-02	●

# LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

## CSTA K-12 - Computer Science Continued...

● = Fully Addresses Benchmark

○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.11
Algorithms and Programming	1A-AP-08	●
	1A-AP-09	●
	1A-AP-10	●
	1A-AP-11	●
	1A-AP-12	●
	1A-AP-13	○
	1A-AP-14	●
	1A-AP-15	●
	1B-AP-09	●
	1B-AP-10	●
	1B-AP-11	●
	1B-AP-12	○
	1B-AP-13	●
	1B-AP-15	●
	2-AP-10	●
	2-AP-11	○
	2-AP-12	●
	2-AP-13	○

# TEACHER GUIDES

Prompts & Questions for Delivery of Lessons

## GUESS THE NUMBER

### WHILE VS FOR LOOPS

In the previous two lessons, students were introduced to two different types of loops. These loops are used in different ways depending on whether you have a definite number of times that the code needs to be repeated. This is when you would use a for loop. Otherwise, if the number of times that the loop is to be repeated depends on a condition being met then you would need to use a while loop. Students will start to compare the two loops and decide which is best based on the context of their programs.

- What are the main differences between for and while loops?
- When do you think you would use a for loop?
- When do you think you would use a while loop?
- What questions could you ask yourself to help decide whether to use a for or while loop?

### PROGRAMMING GUESS THE NUMBER

Students will need to decide what type of loop will be required to create the guess the number game in Python. On top of this, students will need to incorporate Marty's physical features to give the player feedback during the game. For example, they might want to use the arms to suggest whether the number guessed is too high or too low.

- What Marty movements are you going to use to give the player feedback during the game?
- Why have you decided to use these movements/actions?

# POWERPOINT NOTES

Slide Notes to Deliver our PowerPoints

## Lesson 3 Guess the Number

By the end of this lesson you will be able to,

- Describe examples of when a for loop is needed
- Describe examples of when a while loop is needed
- Design and build a program that uses both a for and while loop
- Draw a flowchart for a program with nested loops

Introduction to lesson outcomes

So far...

**FOR loops vs WHILE loops**

What's the difference?

What is the difference between a for and while loop, when have we used them so far?

In pairs, decide which kind of loop would be used in the following examples

Following the workbooks, in pairs students should decide whether a for or while loop would be the best to use based on the context of the program

Scenario 1

Repeatedly asking the user to enter the correct password

**WHILE**

WHILE because you don't know how many times it will take a user to guess the password correctly (you could argue a for loop could be used if the user is only allowed a certain number of guesses)

Scenario 2

Repeating arm movements to recreate a handshake motion

**FOR**

FOR because you know how long you want the handshake to last for so you can decide how many times the arms should go up and down

Scenario 3

Letting the user enter commands a specific number of times

**FOR**

FOR because there is a specific number of times that the user is allowed to enter commands

# POWERPOINT NOTES

Slide Notes to Deliver our PowerPoints

## Scenario 4

Asking the user for input until  
they enter **stop**

**WHILE**

WHILE because we don't know how many times the user will enter a command until they specify STOP

## Live Demonstration

Guess the Number



You might want to play the game a few different times to demonstrate how it works with students

## Over to you...

1. Design a flowchart diagram for guess the number
2. Code your game in Python

Students need to design a flowchart diagram in their workbooks before grabbing a Marty and starting to code

## End of Lesson Reflection

One thing I didn't understand  
One thing I like about my game so far

End of lesson reflection

# SOLUTIONS

Sample Solutions & Activity Guides

## 1 For vs While Loops

Students have been highlighted to both for and while loops using Python and will start to be drawing similarities and differences between the two kinds of loops.

To summarise for loops are generally used when we know that each time the program runs the loop only needs to run a specific number of times whereas a while loop is more flexible in that it keeps running until a condition is met. This allows us to not set a limit on the number of times that the while loop runs as we won't always know the exact number each time the program is run.

Repeatedly asking the user to enter the correct password

**FOR / WHILE**

Repeating arm movements to create a handshake motion

**FOR / WHILE**

Letting the user enter commands a specific number of times

**FOR / WHILE**

Asking the user for input until they enter **stop**

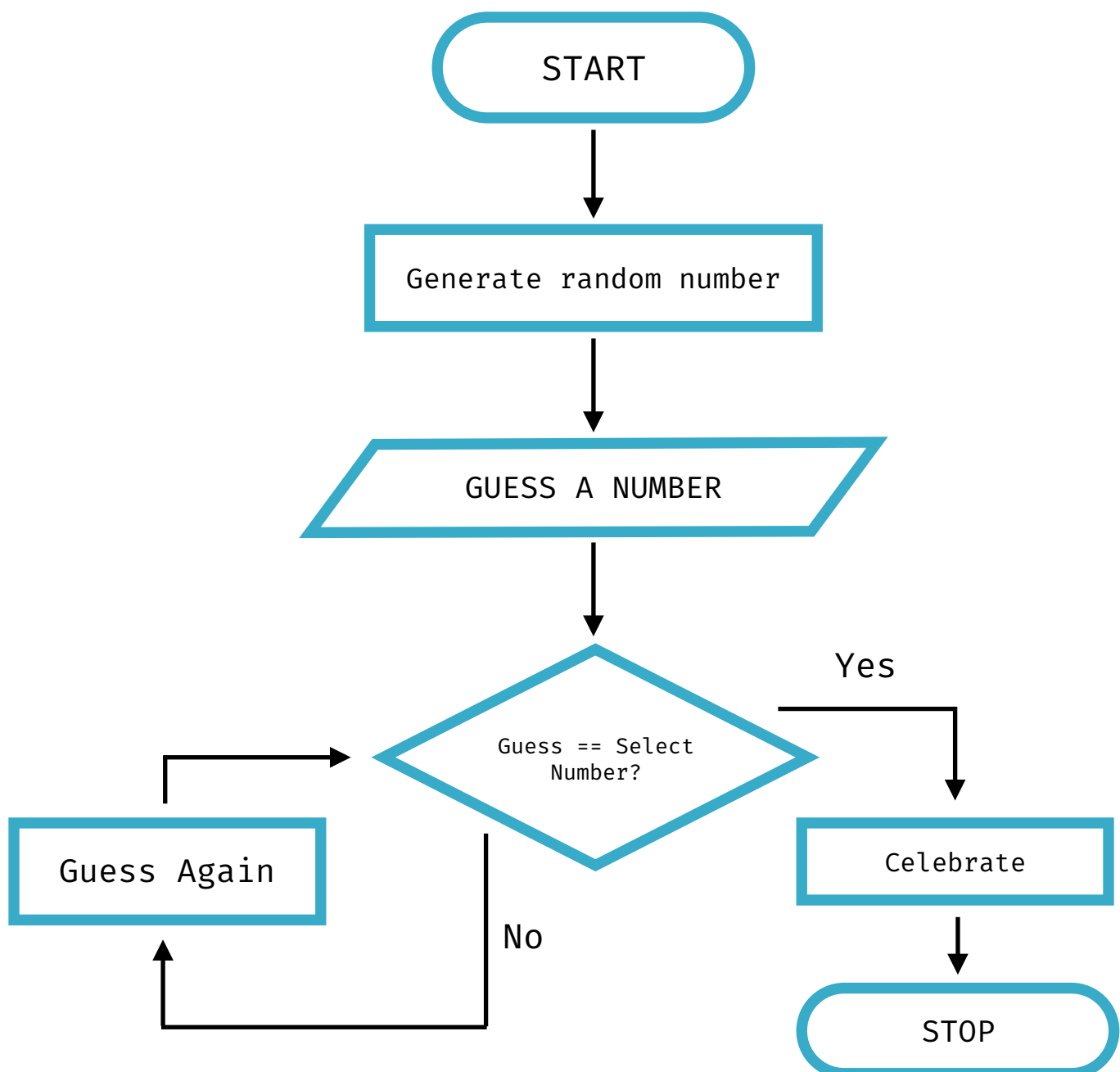
**FOR / WHILE**

# SOLUTIONS

Sample Solutions & Activity Guides

## 2 Guess the Number Flowchart

Here is an example flowchart for the guess the number game. Note that there may be differences between different students depending on how they have decided to do the task but generally it will be similar.





# GUESS THE NUMBER CONTINUED

Outcomes, Resources & Learning Plans

**EDUCATION LEVEL:** Third/Fourth Level (Ages 11-15)

**PRE-REQUISITES:** 3.1, 3.2, 3.5, 3.6, 3.9-3.11

**LESSON DURATION:** 45 minutes

**DEVICE COMPATIBILITY:** Laptop or PC

**CROSS-CURRICULAR LINKS:** Technologies/Literacy/Numeracy

## LESSON OVERVIEW

Continuing from Lesson 1.11, students will finish coding their number guessing game where they will need to use both while and for loops. For students who finish this activity quickly, there are a number of challenges for them to extend their programs with.

### LEARNING OBJECTIVES

- Describe examples of when a for loop is needed
- Describe examples of when a while loop is needed
- Design and build a program that uses both a for and while loop
- Draw a flowchart for a program with nested loops
- Differentiate between a for and while loop

### KEY VOCABULARY

- For Loop
- While Loop
- Number
- Condition
- Control
- User Input

### RESOURCES & EQUIPMENT

- Marty the Robot
- Student workbooks (Lesson 4)
- Devices with a Python editor and *martypy* installed

### ADDITIONAL READING

- Marty the Robot Educator Guide
- Educator FAQ
- Getting Started with Python
- MartyPy Documentation

### LEARNING PLAN & ACTIVITIES

1. Discuss with students how they got on with coding their number guessing game in the previous lesson
  1. What stage is everyone at?
  2. How easy is this task? What has been difficult so far?
2. Students to continue working in their groups to finish coding the guess the number game with Marty and once finished, should work through the challenges below

### EXTENSIONS & CHALLENGES

- What should happen if the player enters in a *string* (text), how should Marty respond? (*Technologies/Literacy/Numeracy*)
- Extend your program so that the player has a limited number of chances to guess correctly, you can decide how many chances the player should get (*Technologies/Numeracy*)
- How could you make your game more difficult? Would changing the range the numbers could be picked from make it more difficult? (*Technologies/Numeracy/Literacy*)
- Add a hint for the player like a fact about the number that has been selected such as if it is an odd or even number (*Technologies/Numeracy*)

# LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

## Curriculum for Excellence - Technologies

● = Fully Addresses Benchmark

○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.12
Digital Literacy	TCH 0-01a	●
	TCH 1-01a	○
	TCH 2-01a	○
Technological Developments in Society and Business	TCH 0-05a	●
Computing Science	TCH 0-13a	●
	TCH 1-13a	●
	TCH 2-13a	●
	TCH 3-13a	●
	TCH 4-13a	○
	TCH 3-13b	●
	TCH 0-14a	●
	TCH 1-14a	●
	TCH 2-14a	●
	TCH 3-14a	○
	TCH 4-14a	○
	TCH 0-14b	●
	TCH 1-14b	●
	TCH 3-14b	●
	TCH 4-14b	●
	TCH 0-15a	●
	TCH 1-15a	●
	TCH 2-15a	●
	TCH 3-15a	●
	TCH 4-15a	○

# LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

## National Curriculum - Computing, Design & Technology

● = Fully Addresses Benchmark

○ = Partially Addresses Benchmark

Curriculum	Organiser	Benchmark Covered	Lesson 3.12
Computing		1-a	●
		1-b	●
		1-c	●
		1-e	●
		2-a	●
		2-b	●
		2-c	●
		3-a	●
		3-b	●
		3-c	●
		3-e	●
		3-g	●
		4-a	●
		4-b	●
Design and Technology		1.1-a	●
		1.3-b	●
		2.3-b	●
		3.3-c	●

# LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

## Australian F-10 Curriculum - Digital Technologies, Design & Technologies

● = Fully Addresses Benchmark

○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.12
Digital Technologies	ACTDIK001	●
	ACTDIK002	●
	ACTDIP004	●
	ACTDIP010	●
	ACTDIP012	○
	ACTDIP013	●
	ACTDIP016	○
	ACTDIP019	●
	ACTDIP028	●
	ACTDIP029	●
	ACTDIP030	●
Design and Technologies	ACTDEK001	●
	ACTDEP005	●
	ACTDEP008	○
	ACTDEP009	●
	ACTDEP018	●
	ACTDEP025	●

# LINKS TO THE CURRICULUM

Support with Benchmarks & Frameworks

## CSTA K-12 - Computer Science

● = Fully Addresses Benchmark

○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 3.12
Computing Systems	1A-CS-01	●
	1A-CS-02	●
	1A-CS-03	○
	1B-CS-01	●
	1B-CS-02	●
	2-CS-02	●
Algorithms & Programming	1A-AP-08	●
	1A-AP-09	●
	1A-AP-10	●
	1A-AP-11	●
	1A-AP-12	●
	1A-AP-14	●
	1A-AP-15	●
	1B-AP-09	●
	1B-AP-10	●
	1B-AP-11	●
	1B-AP-13	●
	1B-AP-15	●
	2-AP-10	●
	2-AP-12	●
	2-AP-13	○
	3A-AP-13	○

# TEACHER GUIDES

Prompts & Questions for Delivery of Lessons

## GUESS THE NUMBER CONTINUED

### REFLECTING ON PROGRESS SO FAR

In the last lesson, students started to code a guess the number game using Python and Marty the Robot. Students should reflect on their progress so far and discuss what problems they have faced whilst coding the game.

- What problems have you come across whilst coding your game?
- How comfortable do you feel explaining for and while loops to someone in the room?
- What is one thing that you do not yet feel comfortable with in Python?

### FURTHER CHALLENGES

Once students have completed the basic functionality of the guess the number game, there are a number of different extensions and challenges they can try to customise their game,

- How easy is it to win your game at the moment? How could you make it more difficult?
- How does your program handle someone entering text instead of a number? How could you solve this problem?
- What comments have others in the class given your game? Have any of these comments given you ideas for extending your game?

# POWERPOINT NOTES

Slide Notes to Deliver our PowerPoints

## Lesson 4 Guess the Number Continued

By the end of this lesson you will be able to,

- Describe examples of when a for loop is needed
- Describe examples of when a while loop is needed
- Design and build a program that uses both a for and while loop
- Draw a flowchart for a program with nested loops
- Differentiate between a for and while loop

Introduction to lesson outcomes

## Guess the Number

What problems are you facing?  
How are you going to extend the game?  
How can you make the game more challenging?

Use the space in your workbooks to keep planning

Reflect on the game students have started to code in the last lesson and discuss problems and ideas for extending the game

## Struggling to think of extension ideas?

- Handling text input from the user instead of numbers
- Only allow the player a limited number of chances to guess correctly
- Try changing the range the random number could be selected from
- Give the player a fact about the select number like if it's an odd/even

Extension ideas

## End of Lesson Reflection

One thing I'm proud of in my game  
One challenge that I tackled

End of lesson reflection

# SOLUTIONS

Sample Solutions & Activity Guides

## 1 Extensions & Planning

Once students have completed the first version of their game, there will be elements of the game that they might want to change or improve by making it more difficult.

There are some suggestions on extension ideas given in the lesson plan and powerpoint slides. Students may choose to try out some of these ideas but may also have some ideas of their own. You might want to discuss with students about what extensions are manageable to complete within the lesson.





