Codepai: Introduction to Blockly, Scratch, Microbit, Robotics and Python.

Unit description	Learning goals	Associated lessons
Unit 1: Get familiar and set up An introduction to Robotics	Students familiarize themselves with the programming environment and how to download a program to the robot. ▪ Set up and become familiar with Edison using barcode programs ▪ Open software and become familiar with how to move icons ▪ Download test program	
Unit 2: Robot movement – driving Introduction to sequential programming	Students learn how the robot responds to command icons and bring together the concepts of time, speed and distance.	

Unit description	Learning goals	Associated lessons
Unit 3: Robot movement – turning Sequential programing and basic geometry	Students learn how the robot responds to time and geometry and how they can achieve driving control of the robot.	Program 1 – Right turn (Worksheet 3.1) Program 2 – Left turn (Worksheet 3.2) Program 3 – Right and then left turn (Worksheet 3.3) Program 4 – Mini maze (Worksheet 3.4)
Lesson 4: Maze challenge and Mexican wave Reinforce learning	Students use knowledge from lessons 1 through 3 to achieve two fun open ended activities.	Program 1– Driving challenge (Worksheet 4.1) Program 2 – Mexican wave robot style (Worksheet 4.2)
Lesson 5: Design brief 1 – My program Creative thinking and problem solving	Students come up with their own challenge and conceptualise how the robot can provide a solution. Students may select their own topic, state the program's purpose and explain where it could be used in the real world. Describe the programming icons used and what they do Demo – Students demonstrate their robot's program to the class	
Lesson 6: Clap sensing Introduction to inputs (sensors)	Students learn how to make the robot respond to outside stimulus (claps). This lesson also includes a fun class activity	Program 1 – Flash LED in response to a clap (Worksheet 6.1) Program 2 – Drive in response to a clap (Worksheet 6.2) Program 3 – Dance in response to clapping (Worksheet 6.3)

Unit description	Learning goals	Associated lessons
Lesson 7: Detect obstacles Introduction to the concept of obstacle detection and artificial intelligence	Students program the robot to make decisions (artificial intelligence) in response to obstacles in the robot's environment. Understanding infrared obstacle detection	Program 1 – Detect an obstacle and stop (Worksheet 7.2) Program 2 – Detect an obstacle and avoid (Worksheet 7.3) Program 3 – Detect an obstacle and avoid in a loop (Worksheet 7.4) Program 4 – Right and left obstacle detection (Worksheet 7.5)
Lesson 8: Line sensing and tracking Industrial like robotic behaviour	Students learn about basic robot sensing and control similar to that used in advanced automated factories and warehouses. Understanding the line tracking sensor	Program 1 – Drive until a black line (Worksheet 8.2) Program 2 – Drive inside a border (Worksheet 8.3) Program 3 – Follow a line (Worksheet 8.4)
Lesson 9: Respond to light Environmental measurement and programming mathematics	Students learn about measuring light levels, storing them in memory and performing mathematics to control the robots behaviour. Understanding variables	Program 1 – Light level alarm (Worksheet 9.2) Program 2 – Automatic lights (Worksheet 9.3) Program 3 – Light following (Worksheet 9.4)

Unit description	Learning goals	Associated lessons
Lesson 10: Design brief 2 – My program Creative thinking and problem solving	This is the second design brief where students come up with their own challenge and conceptualise how the robot can provide a solution 1. Identify a problem that the robot can solve or action it can take 2. Describe the problem or the action the robot needs to make 3. Write the program and test it 4. Failure – The first attempt is never successful. Document it and Keep trying! 5. Describe the programming icons used and what they do 6. Demo – Students demonstrate their robot's program to the class	

1. Introduction to Python, Microbit and Robotics

Unit description	Learning goals	Associated lessons
Unit 1: Talking to your computer An introduction to Python language. Understand basic Python activity and functionality.	▪ Writing your first Python program ▪ Reading user input ▪ String concatenation ▪ Become familiar with the Python programming environment	**Lesson A:** What's a computer and what's your function? **Lesson B:** Introduction to Python
Unit 2: Calculating things An introduction on what Python can do for Maths	▪ Doing Maths with Python ▪ Different types of number in Python ▪ Converting types ▪ Putting it all together!	**Lesson A:** Codepai Online course

Unit description	Learning goals	Associated lessons
Unit 3: Manipulating strings and list in Python Introduce and practice Python basic data structure	▪ Different types of data in Python ▪ Become familiar different Data Structures (list, string) ▪ Understand basic String operation	**Lesson A:** Codepai Online course-jupyter notebook
Unit 4: Manipulating other data structures (dict,tuples, sets) in Python Introduce and practice Python basic data structure	▪ Become familiar different Data Structures (dict, tuples, sets) ▪ Understand basic operation	**Lesson A:** Codepai Online course-jupyter notebook
Unit 5: Python Functions Introduce, define and practice Python basic functions	▪ Become familiar Python functions ▪ Understand the syntax and the usage of Python function ▪	**Lesson A:** Codepai Online course-jupyter notebook : Functions.ipynb
Unit 6: Manipulating Files	▪ Understand the basic process on how to manipulate files ▪ Know how to read and write a file	

Unit description	Learning goals	Associated lessons
Unit 7: Making with micro:bit An introduction to design thinking and the micro:bit as a component for a basic making activity.	▪ Exercise creativity, engineering and resourcefulness by coming up with ideas for using simple household or classroom materials to accommodate the micro:bit's size and weight as part of their micro:pet project. ▪ Test and iterate using different materials and sizes in order to create an optimal design to house the micro:bit and battery pack. ▪ Become familiar with the MakeCode and micropython programming environment. ▪ Learn how to download programs from the computer to the micro:bit.	Lesson A: The micro:bit is for making Lesson B: Introduction to MakeCode and MicroPython Lesson C: micro:pet project
Unit 8: Algorithms Introduces a conceptual framework for thinking of a computing device as something that uses code to process one or more inputs and send them to an output(s).	▪ Understand the four components that make up a computer and their functions. ▪ Understand that the micro:bit takes input, and after processing the input, produces output. ▪ Learn the variety of different types of information the micro:bit takes in as input. ▪ Apply this knowledge by creating a micro:bit program that takes input and produces an output.	Lesson A: What's a computer and what's your function? Lesson B: Happy face, sad face Lesson C: Fidget cube

Unit description	Learning goals	Associated lessons
Unit 9: Variables Introduces the use of variables to store data or the results of mathematical operations, and the importance of giving variables unique and meaningful names.	▪ Understand what variables are, and why and when to use them in a Python program. ▪ Learn how to create a variable, set the variable to an initial value, and change the value of the variable within a micro:bit program. ▪ Learn how to create meaningful and understandable variable names. ▪ Understand that a variable holds one value at a time. ▪ Understand that when you update or change the value held by a variable, the new value replaces the previous value. ▪ Learn how to use the basic mathematical blocks for adding, subtracting, multiplying, and dividing variable values. ▪ Apply the above knowledge and skills to create a unique program that uses variables as an integral part of the project.	Lesson A: Variables in daily life Lesson B: Make a game scorekeeper Lesson C: Everything counts
Unit 10: Conditionals Introduces the Logic blocks, such as 'If…then' and 'If…then…else', with a focus on practicing skills of creativity, problem-solving, and collaboration.	▪ Understand what conditional statements are, and why and when to use them in a program. ▪ Learn how to use the Logic blocks 'If…then' and 'If…then…else'. ▪ Practice using the Logic blocks so different conditions yield specified outcomes. ▪ Demonstrate understanding and apply skill by collaborating with classmates to create a game that uses a micro:bit and a program that correctly and effectively uses conditionals.	Lesson A: Conditionals in daily life Lesson B: Rock, paper, scissors Lesson C: Code a board game

Unit description	Learning goals	Associated lessons
Unit 11: Iteration Introduces the concept of iteration and loops to program repeated sequences of code more efficiently.	▪ Understand the value of iteration in programming. ▪ Understand looping as a form of iteration. ▪ Learn how and when to use the looping blocks 'repeat', 'while', and 'for'. ▪ Apply the above knowledge and skills to create a unique program that uses iteration and looping as an integral part of the project.	Lesson A: Understanding iteration Lesson B: Coding with loops Lesson C: Get loopy
Unit 12: Mini-project Provides a review of the concepts covered in the units 1-5, introduces an independent "mini-project," and reinforces the important idea that programming is a process of patient problem-solving.	▪ Code a unique, original program, and design and build a physical maker component of the project that uses the micro:bit in some way. ▪ Demonstrate the use of one of the following concepts to illustrate what they know and show something new: ▪ Design Thinking ▪ Input / Processing / Output ▪ Variables ▪ Conditional statements ▪ Iteration/loops	Lesson A: Looking back so far Lesson B: Coding and making a mini-project Lesson C: Mini-project showcase

Unit description	Learning goals	Associated lessons
Unit 13: Coordinates Introduces the use of coordinates to store data or the results of mathematical operations, and gives students practice programming the LEDs of the micro:bit screen using coordinates.	▪ Understand that the 5 x 5 grid of LEDs on the micro:bit represents a coordinate grid with the origin (0,0) in the top left corner. ▪ Understand that the values of the x coordinates range from 0 through 4 and increase from left to right. ▪ Understand that the values of the y coordinates range from 0 through 4 and increase from top to bottom. ▪ Learn how to refer to an individual LED by its x and y coordinates. ▪ Learn how to plot (turn on) and unplot (turn off) individual LEDs and how to toggle between these two states. ▪ Learn how to check the current on or off status of an individual LED, as well as check and set the brightness level. ▪ Apply the above knowledge and skills to create a unique program that uses coordinates as an integral part of the project.	Lesson A: Understanding coordinates Lesson B: Coding animations and patterns Lesson C: Screensaver or game project
Unit 14: Booleans Introduces the use of the Boolean data type to control the flow of a program, keep track of state, and to include or exclude certain conditions.	▪ Understand what Booleans and Boolean operators are, and why and when to use them in a program. ▪ Learn how to create a Boolean, set the Boolean to an initial value, and change the value of the Boolean within a micro:bit program. ▪ Learn how to use the random true or false block. ▪ Apply the above knowledge and skills to create a unique program that uses Booleans and Boolean operators as an integral part of the program.	Lesson A: Understanding Booleans Lesson B: Coding a double coin flipper Lesson C: Project Boolean

Unit description	Learning goals	Associated lessons
Unit 15: Bits, bytes, and binary Introduces the concept of binary digits, base-2 notation, how data is stored digitally, and how it can be read and accessed.	▪ Understand what bits and bytes are and how they relate to computers and the way information is processed and stored. ▪ Learn to count in base-2 (binary) and translate numbers from base-10 (decimal) to binary and decimal. ▪ Apply the above knowledge and skills to create a unique program that uses binary counting as an integral part of the program.	Lesson A: Understanding bits, bytes, binary Lesson B: Code a binary transmogrifier Lesson C: Make a binary cash register
Unit 16: Radio communication Introduces the radio functionality of the micro:bit to send and receive numeric and string data between micro:bits, and the concept of pair programming with the project.	▪ Understand how to use the Radio blocks to send and receive data between micro:bits. ▪ Understand the specific types of data that can be sent over the radio. ▪ Work in pairs to apply the above knowledge and skills to design a unique program using radio communication between two micro:bits.	Lesson A: Understanding radio communication Lesson B: Explore the Radio Toolbox Lesson C: Make a micro:bit radio
Unit 17: Arrays Introduces the usefulness of arrays to store a collection of related data types and retrieve the data points in an ordered fashion, and common algorithms for sorting data.	▪ Understand what arrays are, how to create them, and learn common array operations, such as setting and getting values by index. ▪ Explain the steps they would take to sort a series of numbers. ▪ Recognize three common sorting algorithms. ▪ Practice storing and retrieving values in arrays. ▪ Demonstrate understanding and apply skills by creating a musical instrument that uses a micro:bit and a program that correctly and effectively uses arrays to store data.	Lesson A: Understanding arrays Lesson B: Coding with arrays Lesson C: Make a micro:bit musical instrument

Unit description	Learning goals	Associated lessons
Unit 18-20: Independent final project Provides review of units 7-11, and tasks students to create an independent project that demonstrates the use of something they have already learned, something they researched for themselves, something they borrowed from somewhere else (with citations), and something completely original, as well as documentation of their design, making, and learning process.	▪ Code a unique, original program, and design and build a physical maker component of the project that uses the micro:bit in some way. ▪ Demonstrate the use of one of the following concepts to illustrate what they know and show something new: ▪ Coordinates ▪ Booleans ▪ Bits, bytes, and binary ▪ Radio communication ▪ Arrays	Lesson A: Looking back Lesson B: Coding and making a final project Lesson C: Final project showcase