Coding Microbits using Python — Reflections

Module 1: Design & Making with Microbit - Python

This module introduces the microbit as a piece of hardware that has a specific size and weight, and generally must be supported and incorporated as an essential component of a tangible artifact. Focus on making a pet or robot and incorporating the physical microbit as the face of the project.

Module 1: Project MicroRobot

Questions, Interview, and Responses

_____

_____

_____

_____

_____

_____

Sketches MicroRobot

01.2a Name Display Activity

LIst steps to create a microbit program and install it.

01.2b Icons Display Activity

Here are a list of predefined (Image.XXX) icons:

Image.HEART	Image.CLOCK5,	Image.COW
Image.HEART_SMALL	Image.CLOCK4,	Image.MUSIC_CROTCHET
Image.HAPPY	Image.CLOCK3,	Image.MUSIC_QUAVER
Image.SMILE	Image.CLOCK2,	Image.MUSIC_QUAVERS
Image.SAD	Image.CLOCK1	Image.PITCHFORK
Image.CONFUSED	Image.ARROW_N,	Image.XMAS
Image.ANGRY	Image.ARROW_NE,	Image.PACMAN
Image.ASLEEP	Image.ARROW_E,	Image.TARGET
Image.SURPRISED	Image.ARROW_SE,	Image.TSHIRT
Image.SILLY	Image.ARROW_S,	Image.ROLLERSKATE
Image.FABULOUS	Image.ARROW_SW,	Image.DUCK
Image.MEH	Image.ARROW_W,	Image.HOUSE
Image.YES	Image.ARROW_NW	Image.TORTOISE
Image.NO	Image.TRIANGLE	Image.BUTTERFLY
Image.CLOCK12,	Image.TRIANGLE_LEFT	Image.STICKFIGURE
Image.CLOCK11,	Image.CHESSBOARD	Image.GHOST
Image.CLOCK10,	Image.DIAMOND	Image.SWORD
Image.CLOCK9,	Image.DIAMOND_SMALL	Image.GIRAFFE
Image.CLOCK8,	Image.SQUARE	Image.SKULL
Image.CLOCK7,	Image.SQUARE_SMALL	Image.UMBRELLA
Image.CLOCK6,	Image.RABBIT	Image.SNAKE

LIst 2 ways to use the `display._____` command to view the LED display screen.

_____

_____

0.1.2c Icon Animation Activity

Put at least 4 icons together to tell a story. Write each line of the story below.

_____

_____

_____

_____

_____

_____

01.2d Creative Design Activity

Program individual LEDs using the (Sketch LED face designs for your project)

01.3 Project

Microbit Project Faces (Sketch LED face designs for your project)

01 Ideas, Sketches, Planning, Notes, & Reflections —

Coding & Innovation using Microbits - Python

List the steps to create your robot face animation project.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Reflection

Summarize the feedback from your partner. _____

_____

_____

How would you revise your design, if you were to go back and create another?

_____

_____

What was it like designing a project?

_____

_____

Was it a project you enjoyed? Why or why not?

_____

_____

What would you do to redesign the project?

_____

_____

What was it like to interview your partner? What was it like to be listened to?

_____

_____

What was something that surprised you about the process of designing a

micro:project? _____

_____

Describe a difficult point in the process of designing a micro:project and how you

resolved it? __ _____

_____

_____

Rubric

For creative projects such as these, we normally don't use a qualitative rubric to grade the creativity or the match with their partner's needs. We just check to make sure that the micro:project meets the required specifications:

- Program properly downloaded to microbit
- microbit supported so the face is showing
- microbit can be turned on and off without taking critter apart
- Turned in notes on interview process
- Written reflection

Notes

_____

_____

_____

_____

_____

_____

_____

_____

Coding Microbits using Python — Reflections

Module 2: Software & Hardware (Algorithms)

This module 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). Questions to be answered include: What is a computer? What is a microbit and what can it do? Students will be making projects that utilize the sensors and screen output of the micro:bit.

What computers are in your house?

_____

_____

_____

What are the parts of a computer? List examples of each.

Blackbox - What are the parts in a Blackbox?

$\Rightarrow$		$\Rightarrow$

What are some examples of Blackboxes in society?

_____

_____

_____

_____

02.2a Sensors Temperature Activity

Pseudocode for Sensors Temperature program

_____

_____

_____

_____

_____

Button A Press Event

What is the code for a button A pressed event?

_____

_____

_____

_____

_____

What comments should be included at the top of a program?

_____

_____

_____

_____

How could your microbit be calibrated to reflect the current room temperature?

_____

What are some other **sensor** inputs on the microbit?

_____

_____

02.2b Sensors Temperature & Compass Activity

Add a button B pressed event to your Sensor code to find the compass direction. Pseudocode to add a compass to the Sensors Temperature program

_____

_____

_____

_____

_____

02.2c Accelerometer Tilt Activity

Code different tilt events and display an arrow or a word showing the tilt direction. Pseudocode to create a tilt events program. (See Python Microbit Notes)

_____

_____

_____

_____

_____

_____

_____

02.3 Project: Blackbox

In this project you will plan, design, and create your own Blackbox using different events, microbit sensors, output to LEDs, and MakeCode's block programming. You will also use a maker elements as part of you design and construction.
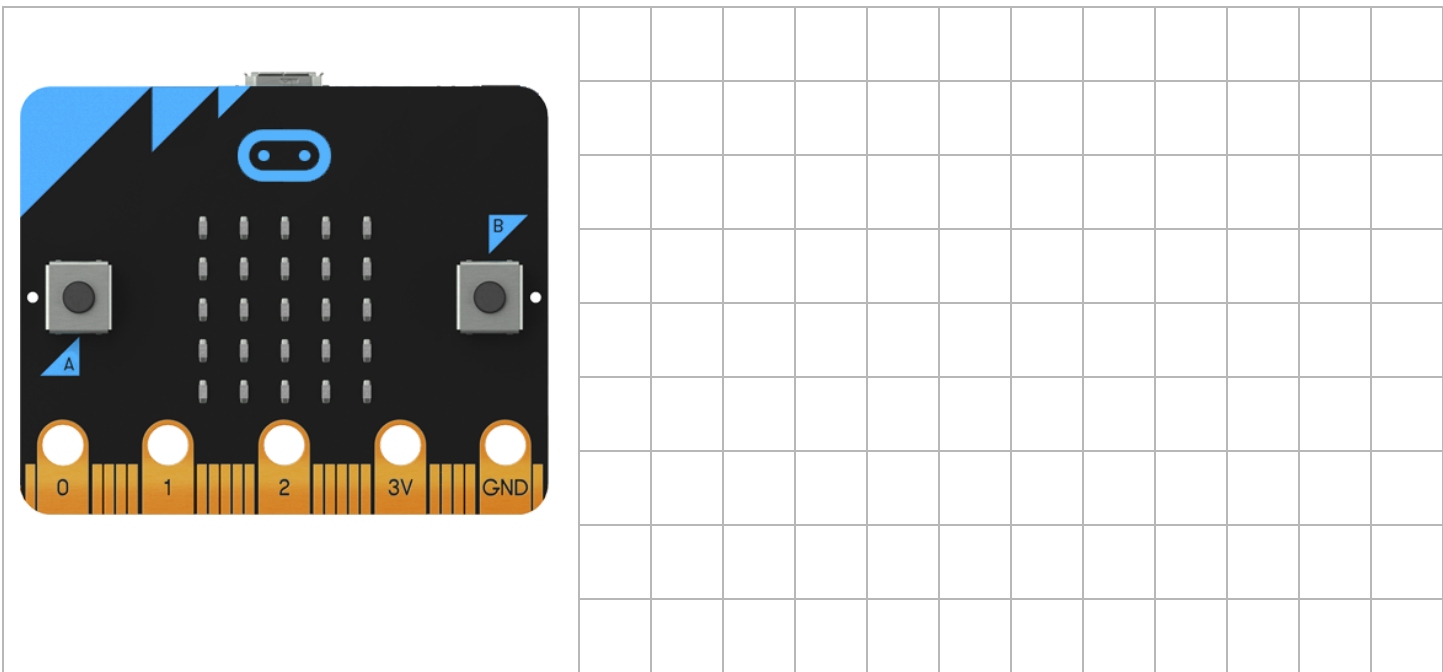
Brainstorm Ideas _____

_____

_____

_____

Project: _____

Description: _____

_____

Project Sketch:

Blackbox Algorithm & Pseudocode

_____

_____

_____

_____

_____

_____

_____

_____

_____

Materials Needed: _____

_____

Coding Plan: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Notes & Reflections

Beta Testing: _____

_____

Revision Ideas: _____

_____

Photos:

Assessment Rubric - Competency scores

Competencies	4	3	2	1
Inputs	At least 4 different inputs are successfully implemented.	At least 3 different inputs are successfully implemented.	At least 2 different inputs are successfully implemented.	Fewer than 2 different inputs are successfully implemented.
Outputs	At least 4 different outputs are successfully implemented.	At least 3 different outputs are successfully implemented.	At least 2 different outputs are successfully implemented.	Fewer than 2 different outputs are successfully implemented.
Micro:bit Program	micro:bit program: 1) uses event handlers in a way that is integral to the program 2) compiles and runs as intended 3) includes meaningful comments	micro:bit program lacks 1 of the required elements.	micro:bit program lacks 2 of the required elements.	micro:bit program lacks all of the required elements.
Collaboration Reflection	Reflection piece includes: 1) brainstorming ideas 2) construction 3) programming 4) beta testing	Reflection piece lacks 1 of the required elements.	Reflection piece lacks 2 of the required elements.	Reflection piece lacks 3 of the required elements.

Notes

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Python Microbits Code Notes

Events Code

Here is a link to all the possible button and gesture events associated with the microbit. Any section(s) can be copied and used. This code is meant to make it is easier to get started using events on the microbit.

```python
# 2.0 Event Structures
# by C Lyman
# July 2019
# Module 2 of Coding & Innovation using Microbits - Python
# Structures for different events using Microbits

from microbit import *

# forever loop for Events
while True:
    # Event - button A pressed?
    if button_a.is_pressed():
        # action when A is pressed
        display.show("A")

    # Event - button B pressed?
    if  button_b.is_pressed():
        # action when B is pressed
        display.show("B")

    # Event - buttons AB pressed?
    if  button_a.was_pressed() and button_b.was_pressed():
        # action when A&B are pressed
        display.scroll("AB")

    # Event - pin0 touched?
    if  pin0.is_touched():
        # action when pin0 & ground are touched
        display.show("0")

    # Event - pin1 touched?
    if  pin1.is_touched():
        # action when pin1 & ground are touched
        display.show("1")

    # Event - pin2 touched?
    if  pin2.is_touched():
        # action when pin2 & ground are touched
        display.show("2")

    # Event gesture face up
    faceUp = accelerometer.was_gesture("face up")
        if faceUp:
        display.scroll("UP")
```

```python
# Event gesture face down
    faceDown = accelerometer.was_gesture("face down")
    if faceDown:
        display.scroll("DWN")

    # Event gesture shake
    shake = accelerometer.was_gesture("shake")
    if shake:
        display.scroll("SHK")

    # Event gesture up
    up = accelerometer.was_gesture("up")
    if up:
        display.scroll("^")

    # Event gesture down
    down = accelerometer.was_gesture("down")
    if down:
        display.show("v")

    # Event gesture right
    right = accelerometer.was_gesture("right")
    if right:
        display.show(">")

    # Event gesture left
    left = accelerometer.was_gesture("left")
    if left:
        display.show("<")

    # Event - freefall?
    freefall = accelerometer.was_gesture("freefall")
    if freefall:
        # action when microbit is in freefall
        display.scroll("FF")

    # Event - 3g?
    threeG = accelerometer.was_gesture("3g")
    if threeG:
        #6g & 8g are also options
        # action when microbit is accelerated at 3G
        display.scroll("3G")
```

Sensors Code

There a many sensors that can be used as input from the microbit. Below are snippets of code that can be used to access the sensors using MicroPython. For the most part they return a numeric value that can be displayed on the LED screen or used in a calculation.

```python
# 2.0 Sensors
# by C Lyman
# April 2019
# Module 2 of Coding & Innovation using Microbits - Python
# Code for different sensors using Microbits

from microbit import *

while True:
# Sensor code lines and examples with value stored in a variable

# Temperature in Celsius
# Code: temperature()
# Example:
    temp = temperature()

# Light level from the display 0-255
# Code: display.read_light_level()
# Example:
    light = display.read_light_level()

# Acceleration x - tilting left - right +
# 0 when flat facing up
# Code: accelerometer.get_x()
# Example:
    accelX = accelerometer.get_x()

# Acceleration y - tilting forward + back -
# 0 when flat facing up
# Code: accelerometer.get_y()
# Example:
    accelY = accelerometer.get_y()

# Acceleration z - moving up + down -
# -1024 when flat face up (Gravity acting downwards)
# 1024 when face down
# vigorous movement will get values +-2048
# Code: accelerometer.get_z()
# Example:
    accelZ = accelerometer.get_z()

# Acceleration all axes
# Code: accelerometer.get_values()
# Example:
    accelXYZ = accelerometer.get_values()
```

```python
# Compass Calibrate
# The compass must be calibrated before it can be used.
# The microbit asks you to "tilt until the screen is filled".
# When that is completed a smiley face shows on the screen.
# Then the compass will work.
# Code: compass.calibrate()
# Example:
    compass.calibrate()


# Compass Heading
# Gives a compass degrees for the direction top of the microbit
# (away from the pins) is pointed. 0 or 360 North, 90 East,
# 180 South, and 270 West.
# Code: compass.heading()
# Example:
    compassHeading = compass.heading()


# Compass x gives a magnetic field strength reading in nano tesla
# Code: compass.get_x()
# Example:
    magnetismX = compass.get_x()


# Compass Strength gives an indication of the magnitude
# of the magnetic field strength around the device in nano tesla
# Code: compass.get_field_strength()
# Example:
    magnetismStrength = compass.get_field_strength()
```

Coding Microbits using Python — Reflections

Module 3: Everything Counts (Variables)

Computer programs process information. Some of the information that is input, stored, and used in a computer program as values that vary or change during the running of a program. Programmers create variables to hold the value of information that may change. In a game program, a variable may be created to hold the player's current score, since that value would change (hopefully!) during the course of the game. Students will be making projects like a people counter, pedometer, score keeper, and/or dice roll.

Module 3: Everything Counts (Variables)
List constants and variables in your life.

_____

_____

_____

Kinds of variables (list examples):

number: _____

string: _____

boolean: _____

list: _____

Play **Newspaper Toss** with a team:

Game 1	Score
Team 1	
Team 2	

Game 2	Score
Team 1	
Team 2	

Rules for naming variables and identifiers:

- Use descriptive names
- Start with lowercase letters
- Only use letters (a-z) and numbers (0-9). **No** spaces or symbols
- Use camelCase when putting 2 words together
- An underscore "_" can also be used to connect words
- Constants are done in all CAPS
- Math operators: +, -, *, /, %, **, //

What is the assignment operator and how does it work?

_____

03.2a People Counter Activity

Pseudocode to create a People Counter program.

_____

_____

_____

_____

_____

03.2b Score Keeper Activity

Pseudocode to create a Score Keeper program.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Modifications to People Counter or Score Keeper programs

_____

_____

_____

_____

03.3 Project: Everything Counts

In this project you will plan, design, and create something that counts. It should keep track of input by storing values in variables, process the input, and output in some visual and useful way. It should also perform mathematical operation on the variables to give useful output. It should also use a maker elements as part of the design and construction.
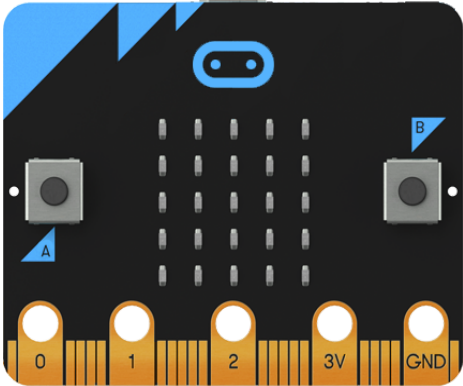
Brainstorm Ideas _____

_____

_____

_____

_____

Project: _____

Description: _____

_____

Project Sketch:

Everything Counts Algorithm & Pseudocode

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Materials Needed: _____

_____

_____

_____

Everything Counts Algorithm & Pseudocode

Coding Plan: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Notes & Reflections

What problem were you trying to solve? _____

_____

How well did your prototype work? _____

_____

What did you change? _____

_____

Describe a difficult point and how you resolved it: _____

_____

_____

Photos

Assessment Rubric - Competency scores

Competency	4	3	2	1
Variables	At least 3 different variables are implemented in a meaningful way.	At least 2 variables are implemented in a meaningful way.	At least 1 variable is implemented in a meaningful way.	No variables are implemented.
Variable Names	All variable names are unique and clearly describe what information values the variables hold using CamelCase	The majority of variable names are unique and clearly describe what information values the variables hold.	A minority of variable names are unique and clearly describe what information values the variables hold.	None of the variable names clearly describe what information values the variables hold.
Mathematical Operations	Uses a mathematical operation on at least two variables in a way that is integral to the program.	Uses a mathematical operation on at least one variable in a way that is integral to the program.	Uses a mathematical operation incorrectly or not in a way that is integral to the program.	No mathematical operations are used.
Micro:bit Program	micro:bit program: 1) Uses variables in a way that is integral to the program 2) Uses mathematical operations to add, subtract, multiply, and/or divide variables 3) Compiles and runs as intended 4) Meaningful comments in code.	micro:bit program lacks 1 of the required elements.	micro:bit program lacks 2 of the required element.s	micro:bit program lacks 3 or more of the required elements.
Collaboration Reflection	Reflection piece addresses all prompts.	Reflection piece lacks 1 of the required elements.	Reflection piece lacks 2 of the required elements.	Reflection piece lacks 3 of the required elements.

Notes

Coding Microbits using Python — Reflections

Module 4: Making Decisions (Conditionals)

Computer programs are instructions telling the computer how to process input and deliver output. An important part of programming is telling the computer WHEN to perform a certain task. For this, we use something called 'conditionals'. Conditionals get their name because a certain Condition or Rule has to be met. Conditionals are usually implemented using an 'if (condition) then action statement. Students will be creating and making projects like coin toss, Magic 8 Ball, and/or dice toss with dots instead of numbers.

Module 4: Making Decisions (Conditionals)
List 3 decisions you have made today.

_____

_____

_____

Conditionals:

if (*condition*) then

> *Action if true*

else

> *Action if condition is false*

Red Light - Green Light conditionals:

if (_____) then

> _____

if (_____) then

> _____

if (_____) then

> _____

else

> _____

Conditions and Boolean expressions

All of the conditions in an if...then statement have to be an expression that can be evaluated as either True or False. These are called Boolean expressions when they are either True or False. These expressions usually use comparison operators to decide if it is True or False.

Comparison Operators		
Operator	Name	Example
==	equals	`x == y`
!=	not equal	`x != y`
>	greater than	`x > y`
<	less than	`x < y`
>=	greater than or equal to	`x >= y`
<=	less than or equal to	`x <= y`

Logical Operators		
Operator	Description	Example
and	Returns True if both statements are true	x < 5 and x < 10
or	Returns True if one of the statements are true	x < 5 or x < 4
not	Reverse the result, returns False if the result is true	not(x < 5 and x < 10)

04.2a Coin Toss Activity

Algorithm & Pseudocode:

_____

_____

_____

_____

_____

_____

_____

_____

_____

04.2b Dice Roll Activity

Algorithm & Pseudocode:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

04.3 Project: Board Game (done with a partner)

In this project you will plan, design, and create a board game. It should have clear rules on how to play. It should use conditionals on the microbit in a way that is central to the game. It should also use a maker elements as part of the design and construction. (Do a search for "DIY board games.)

Brainstorm Ideas _____
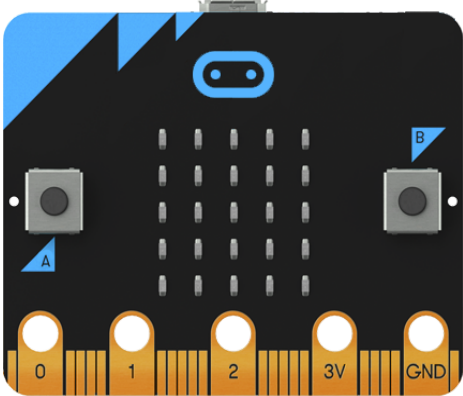
_____

_____

_____

Project: _____

Description: _____

_____

Microbit Project Sketch:

Game Board Sketches

Board Game Rules and Conditionals in Playing:

_____

_____

_____

_____

_____

_____

_____

Board Game Algorithm & Pseudocode:

_____

_____

_____

_____

_____

_____

_____

_____

_____

Materials Needed: _____

_____

_____

Coding Plan: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Photos:

Notes & Reflections

How did you decide with your partner on your board game? _____

_____

_____

_____

What was something that was surprising to you in the creation of your board game?

_____

_____

_____

How well did your prototype work? _____

_____

Describe a difficult point in designing your game and how you resolved it:

_____

_____

_____

What feedback did you get from your beta testers?

_____

_____

What did you change to improve your game? _____

_____

Describe how you and your partner shared the work on the project.

_____

_____

_____

Assessment Rubric

Competency scores

Competency	4	3	2	1
Rules	All game rules are clear and complete.	A game rule is missing or not complete or not clear.	More than one game rule is missing or not complete or not clear.	Most of the game rules are missing or it is not clear what the rules are.
Game Board	Game board is: 1) Complete 2) Neat 3) Fits with the theme of the game 4) micro:bit is a central part of the game	Game board meets only 3 of the conditions listed for a score of 4.	Game board meets only 2 of the conditions listed for a score of 4.	Game board meets only 1 of the conditions listed for a score of 4.
Micro:bit Program	micro:bit program: 1) Uses the micro:bit in a way that is integral to the game 2) Uses conditionals correctly 3) Compiles and runs as intended 4) JavaScript includes comments in code	micro:bit program lacks 1 of the required elements.	micro:bit program lacks 2 of the required elements.	micro:bit program lacks 3 of the required elements.
Photo Documentation	Complete photo documentation that includes photos of game board and code and captions.	A photo is missing or of poor quality or a caption is missing.	Multiple photos and/or captions missing or of poor quality.	Most photos and/or captions missing or of poor quality.
Collaboration Reflection	Reflection piece includes: 1) Brainstorming ideas 2) Construction 3) Programming 4) Beta testing	Reflection piece lacks 1 of the required elements.	Reflection piece lacks 2 of the required elements.	Reflection piece lacks 3 of the required elements.

Comments or Photos:

04 Ideas, Sketches, Planning, Notes, & Reflections —

Coding & Innovation using Microbits - Python

Notes

Coding Microbits using Python — Reflections

Module 5: Music, Designs & LEDs (Loops)

One of the things that computer are really good at is doing the same thing over and over again without getting tired. In this module students will use loops to compose music, connect headphones/speaker, connect LED's, and designs that repeat.

Module 5: Music, Designs, & LEDs (Loops)
List 3 things you did today that involved loops.

_____

_____

_____

Loops

for i in range (*number*):
     do *Actions_____* .

while (*condition*):
     do *Actions if condition is true* _____ .

for i in range (*start number, end number, increment number*):
     do *Actions* _____ .

i is a variable that is the index for the loop.

Repeat loops activities

05.2a Heart Beat Activity

Algorithm & Pseudocode -**for i in range()** loop

_____

_____

_____

_____

_____

_____

05.2b Music Tunes Activity

Algorithm & Pseudocode - **for i in range()** loop

_____

_____

_____

_____

_____

Modification: add a button B pressed for additional tunes.

05.2c Frere Jacques Song Activity

Algorithm & Pseudocode - **for i in range()** loop

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Musical Notes
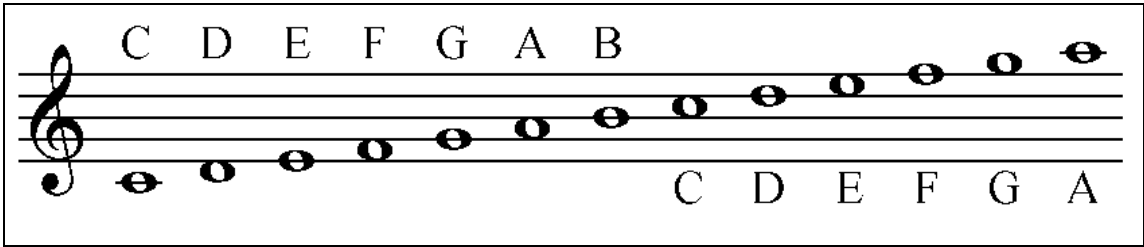
Note values

Modifications for Frere Jacques Song activity. Find another song and add the code and notes to a button B pressed. Create a song of your own and code it.

Algorithm & Pseudocode

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

05.2d European Siren Activity

Algorithm & Pseudocode - **while (*condition*)** loop

_____

_____

_____

_____

_____

_____

05.23e External LED Activity

Algorithm & Pseudocode - **for i in range()** loop

_____

_____

_____

_____

_____

_____

How do you add & program LEDs? _____

_____

_____

_____

05.2g Counting Numbers Activity

Algorithm & Pseudocode - **for i in range(*start, end, increment*)** loop

_____

_____

_____

_____

_____

Modifications: Count by a different number than 1. Count backwards.

Why do some coders just use "i" for the index variable?

_____

_____

What are different ways to increment a variable?

_____

05.3 Project: Loopy Entertainment and Innovation!

In this project you will plan, design, and create an entertaining microbit program that uses loops. It should use at least different 3 loops. It should use sound, display, and/or motion in a way that is central to the project. It should also use a maker elements as part of the design and construction.
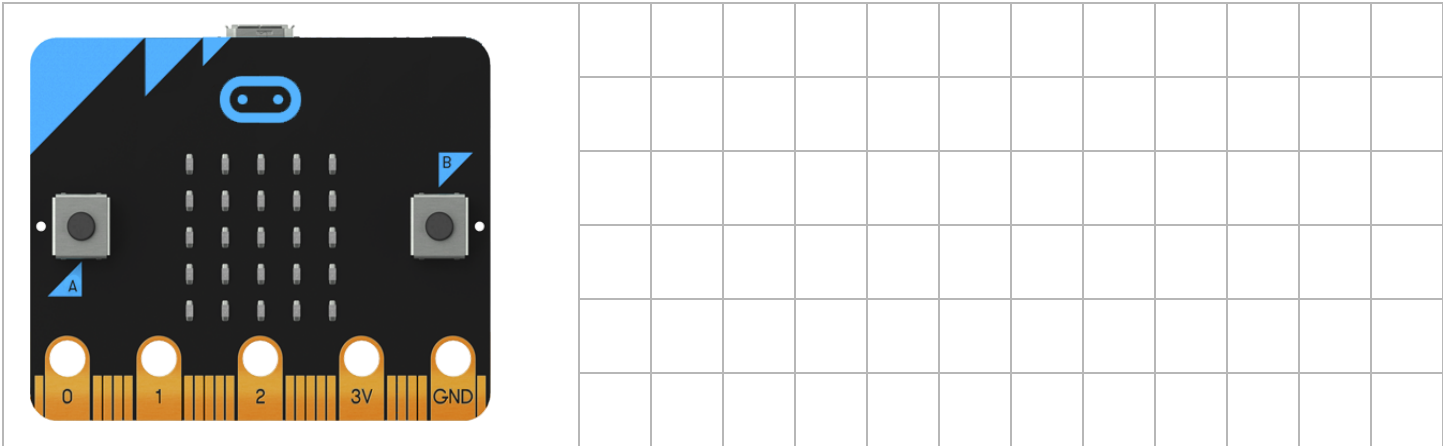
Brainstorm Ideas _____

_____

_____

_____

Project: _____

Description: _____

_____

Microbit Project Sketch:

Entertainment Loopy Sketches

Loopy Entertainment Algorithm & Pseudocode:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Materials Needed: _____

_____

_____

Coding Plan: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Photos:

Photos:

05 Ideas, Sketches, Planning, Notes, & Reflections —
Coding & Innovation using Microbits - Python

Notes & Reflections

How did you decide with you decide on your loopy entertainment?

_____

_____

_____

What was something that was surprising to you in the creation of your project?

_____

_____

_____

How well did your prototype work? _____

_____

Describe a difficult point in designing your loopy entertainment and how you resolved it:

_____

_____

_____

What feedback did you get from your beta testers?

_____

_____

What did you change to improve your loop demo? _____

_____

Assessment Rubric

Competency scores

Competency	4	3	2	1
Loops	At least 3 different loops are implemented in a meaningful way.	At least 2 loops are implemented in a meaningful way.	At least 1 loop is implemented in a meaningful way.	No variables are implemented.
Variables (parameters)	All variable names are unique and clearly describe what information values the variables hold.	The majority of variable names are unique and clearly describe what information values the variables hold.	Few variable names are unique or clearly describe what information values the variables hold.	None of the variable names clearly describe what information values the variables hold.
Sound, Display, & Motion	Uses sound, display, and motion in a way that is integral to the program.	Uses only two of the required elements in a way that is integral to the program.	Uses only one of the required elements in a way that is integral to the program.	None of the required elements are used.
Micro:bit Program	micro:bit program: 1) Uses loops in a way that is integral to the program 2) Compiles and runs as intended 3) Meaningful comments in code	micro:bit program lacks 1 of the required elements.	micro:bit program lacks 2 of the required elements.	micro:bit program lacks 3 or more of the required elements.
Collaboration Reflection	Reflection piece includes: 1) Brainstorming ideas 2) Construction 3) Programming 4) Beta testing	Reflection piece lacks 1 of the required elements.	Reflection piece lacks 2 of the required elements.	Reflection piece lacks 3 of the required elements.

Notes

_____

_____

_____

_____

_____

_____

_____

Coding Microbits using Python — Reflections

Module 6: Radio Communications

This module covers the use of more than one micro:bit to share and combine data. Students will explore the Radio functionality of the micro:bit. Students will send and receive numbers and strings in a series of guided activities. Finally, students are asked to collaborate so that they can share their micro:bits and create a project together that uses the radios as part of their project.

Radio Communications
List all of the different kinds of communication that you can think of that use radios.
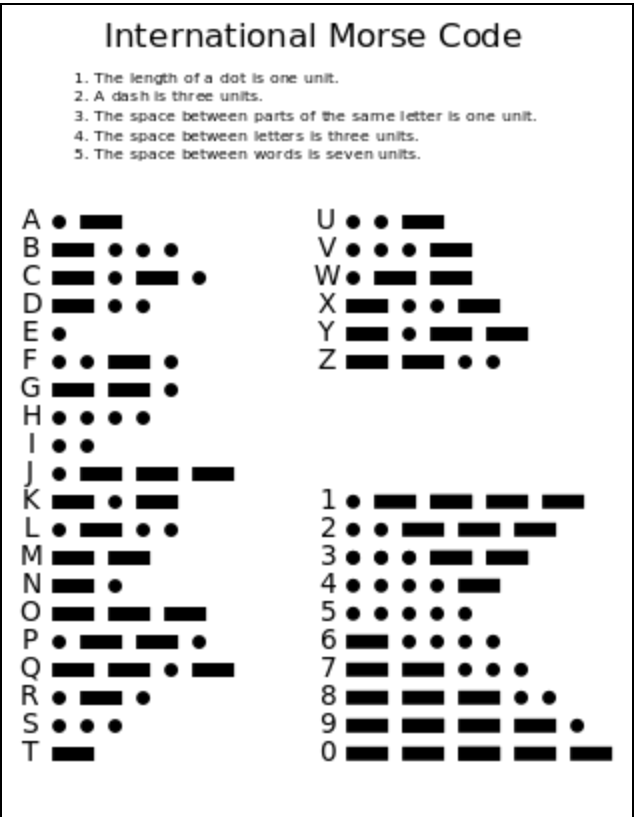
_____

_____

_____

Vocabulary

transmitter _____

receiver _____

Morse Code:

Morse code was one of the first kinds of communication that could be used over a long distance. The first commercial electrical telegraph was developed in May 1837 in London. In the USA Samuel Morse independently developed a telegraph along with the Morse code and sent the first message in January 1838.

At the end of 1894, the young Italian inventor, Guglielmo Marconi, begin working with wireless radio. In March 1897, Marconi transmitted Morse code over a distance of 6 km (3.7 miles). Morse code continued to be used for communication until the 1990s. At one time all Boy Scouts had to learn Morse Code to earn the 1st Class rank.
(Telegraphy. Wikipedia. https://en.wikipedia.org/wiki/Telegraphy)
(Morse Code. Wikipedia. https://en.wikipedia.org/wiki/Morse_code)

International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.

Write your name in Morse Code.

_____

Write a message in Morse Code. Have your neighbor decode it.

_____

Decode your neighbor's message.

_____

06.2a Radio Initials Activity

Algorithm & Pseudocode:

_____

_____

_____

_____

_____

_____

_____

_____

06.2b Morse Code Activity

Algorithm & Pseudocode:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Morse Code Modifications:

_____

_____

_____

_____

_____

06.3 Project: Radio Communications (done with a partner)
In this project you will plan, design, and create a microbit program that uses radio communications. It may have 2 separate programs: one that sends data and the other that receives data. It should also use a maker elements as part of the design and construction.
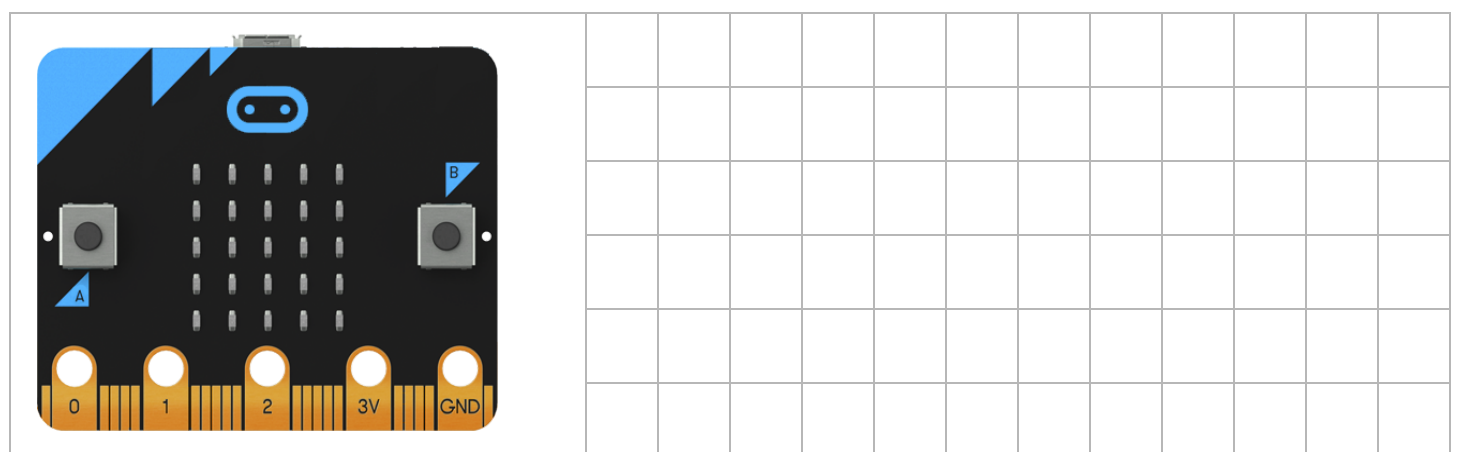
Brainstorm Ideas _____

_____

_____

_____

_____

Project: _____

Description: _____

_____
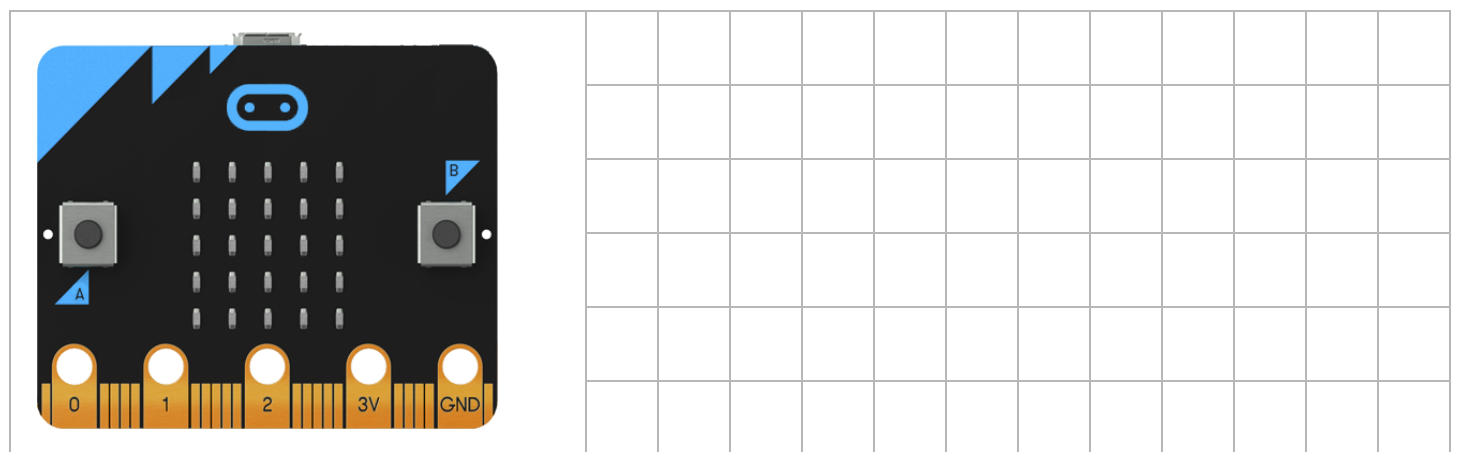
Microbit Project Sketch 1:

Microbit Project Sketch 2:

Radio Communication Sketches

Radio Communications Algorithm & Pseudocode:

_____

_____

_____

_____

_____

_____

_____

_____

_____

06 Ideas, Sketches, Planning, Notes, & Reflections —
Coding & Innovation using Microbits - Python

Materials Needed: _____

_____

_____

Coding Plan: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Photos:

Notes & Reflections

How did you decide with you decide on your radio communication project?

_____

_____

_____

How did your project use radio communications?

_____

_____

_____

Describe something in your project that you are proud of?

_____

_____

_____

What was something that was difficult in the creation of your project?

_____

_____

_____

How well did your prototype work? _____

_____

What feedback did you get from your beta testers?

_____

_____

What did you change to improve your project? _____

_____

Describe how you and your partner shared the work on the project.

_____

_____

_____

_____

Assessment Rubric

Competency scores

Competency	4	3	2	1
Radio	Effectively uses the Radio to send and receive data, with meaningful actions and responses for each.	Effectively uses the Radio to send or receive data, with meaningful actions and responses for each.	Use of Radio is incomplete or non-functional and/or tangential to operation of program.	No working and/or meaningful use of Radio.
Micro:bit Program	micro:bit program: 1) Uses Radio blocks in a way that is integral to the program 2) Compiles and runs as intended 3) Meaningful comments in code	micro:bit program lacks 1 of the required elements.	micro:bit program lacks 2 of the required elements.	micro:bit program lacks all of the required elements.
Collaboration Reflection	Reflection piece addresses all prompts.	Reflection piece lacks 1 of the required elements.	Reflection piece lacks 2 of the required elements.	Reflection piece lacks 3 of the required elements

Notes

_____

_____

_____

_____

_____

_____

_____

Coding Microbits using Python — Reflections

Module 7: Innovation & Mini-Project

The mini-project will give you an opportunity to plan, design, and create a microbit program that serves a purpose by solving a problem or filling a need. It is an opportunity to show what you have learned and to learn something new. It should also use maker elements as part of the design and construction.
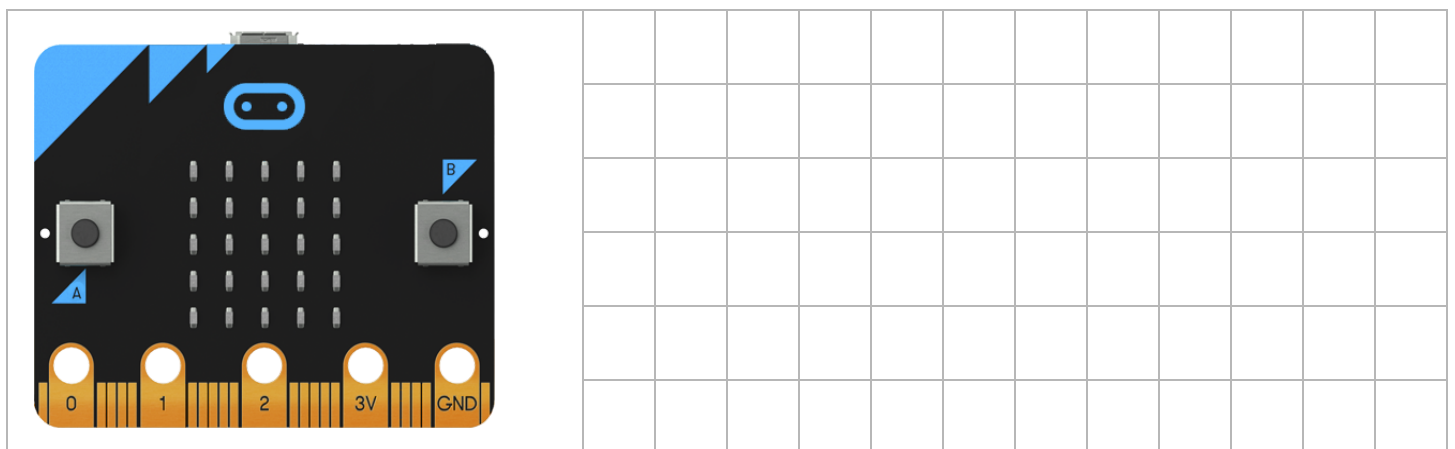
Brainstorm Ideas _____

_____

_____

_____
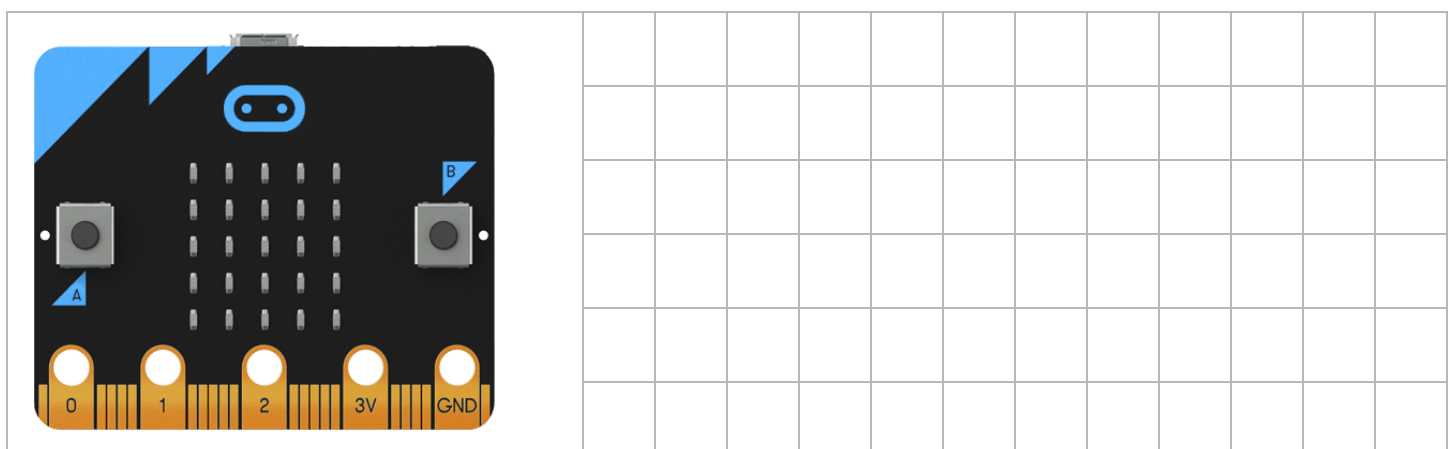
Project: _____

Description: _____

_____

Microbit Project Sketch 1:

Microbit Project Sketch 2:

Mini-project Sketches

Mini-project Sketches

Mini-project Algorithm & Pseudocode:

_____

_____

_____

_____

_____

_____

_____

_____

_____

Materials Needed: _____

_____

_____

Coding Plan: _____

_____

_____

_____

_____

_____

_____

_____

_____

07 Ideas, Sketches, Planning, Notes, & Reflections —

Coding & Innovation using Microbits - Python

_____

Mini-Project Daily Work Log:

Date: _____ Project Goal: _____

_____ _____

_____ _____

_____ _____

Date: _____ Project Goal: _____

_____ _____

_____ _____

_____ _____

Date: _____ Project Goal: _____

_____ _____

_____ _____

_____ _____

_____ _____

Date: _____ Project Goal: _____

_____ _____

_____ _____

_____ _____

_____ _____

Date: _____ Project Goal: _____

_____ _____

_____ _____

_____ _____

_____ _____

Date: _____ Project Goal: _____

_____ _____

07 Ideas, Sketches, Planning, Notes, & Reflections —
Coding & Innovation using Microbits - Python

_____

_____	_____
_____	_____
_____	_____

Photos:

Notes & Reflections

Talk about one challenge you faced in creating this project? How did you overcome it?

_____

_____

_____

_____

How did you demonstrate what you already knew?

_____

_____

_____

What was a new thing you learned? How did you learn it?

_____

_____

_____

Who in the class provided help for you along the way? How?

_____

Describe one specific thing you are proud of in this project.

_____

_____

If you had more days, what would you do to improve your project?

_____

_____

What did you do to publicly show the project?

_____

_____

Assessment Rubric

Competency scores

Competency	4	3	2	1
Code - Show what you know	Code very effectively demonstrates the use of previous concept(s). Variable names are unique and clearly describe what information values the variables hold. Code is highly efficient. Code is commented.	Code only partially demonstrates previous concepts, and/or is not efficient.	Code only partially demonstrates previous concepts, and/or is not efficient, variable names not clear.	Code does not demonstrate previous concepts, is not efficient, variable names not clear.
Code - Show something new	Code very effectively demonstrates the use of new concept(s). Variable names are unique and clearly describe what information values the variables hold. Code is highly efficient. Code is commented.	Code only minimally demonstrates new concepts, and/or is not efficient.	Code only minimally demonstrates new concepts, and/or is not efficient, variable names not clear.	Code does not demonstrate new concepts, is not efficient, variable names not clear.
Maker Component	Tangible component is tightly integrated with the micro:bit and each relies heavily on the other to make the project complete.	Tangible component is somewhat integrated with the micro:bit but is not essential.	Tangible component does not add to the functionality of the program.	No tangible component.
Work Logs	All work logs submitted on time, and accurate.	One late or missing work log and/or work logs not accurate nor sufficiently detailed.	Two late or missing work logs and/or work logs not accurate nor sufficiently detailed.	More than two late or missing work logs and/or not accurate nor sufficiently detailed.
Reflection	Reflection piece describes: 1) Development Process 2) Something new 3) Something proud of 4) Future modifications	Reflection piece lacks 1 of the required elements.	Reflection piece lacks 2 of the required elements.	Reflection piece lacks 3 of the required elements.

Notes & Sketches

07 Ideas, Sketches, Planning, Notes, & Reflections —
Coding & Innovation using Microbits - Python

Notes