Microsoft Intro to CS Curriculum Overview Table

Unit	Unit Overview	Learning Goals	Lesson	Lesson Plan
1. Making with micro:bit	introduces the micro:bit and explains how it can be used. The focus is on incorporating the physical micro:bit into 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 programming environment. ▪ Learn how to download programs from the computer to the micro:bit. ▪ Exercise communication and collaboration and apply the design thinking process to develop an understanding of a problem or user need and iteratively design an optimal solution.	A. The micro:bit is for making	1. Introduce the concept of making and coding. 2. Lead the design thinking unplugged activity.
			B. Introduction to MakeCode	1. Lead the Introduction to MakeCode activity. 2. Assess learning with the unit quiz.
			C. micro:pet project	Introduce and facilitate the project.

		▪ Apply their understanding of the problem in a creative way by making a "micro:pet" creature for their partner.		
2. 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.	A. How computers function	1. Introduce the four main components that make up any computer. 2. Lead the two unplugged activities about functions and conditionals.
			B. Code with event handlers	1. Lead the "Happy face, sad face" coding activity. 2. Assess learning with the unit quiz.
			C. Fidget cube	Introduce and facilitate the coding project.
3. Variables	Introduces the use of variables to store data or the results of mathematical	▪ Understand what variables are and why and when to use them in a program. ▪ Learn how to create a variable, set the variable to	A. Understanding variables	1. Introduce the concept of variables. 2. Lead the "Keeping score" unplugged activity.

	operations and the importance of giving variables unique and meaningful names.	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.	B. Making a game scorekeeper	1. Lead the "Scorekeeper" activity. 2. Assess learning with the unit quiz.
			C. Everything counts	Introduce and facilitate the project.
4. Conditionals	Introduces the Logic blocks, such as 'If…then' and 'If…then…else',	▪ Understand what conditional statements are and why and when to use them in a program.	A. Conditionals in daily life	1. Introduce the concept of conditionals. 2. Lead the "Red light, green light" unplugged activity.

	with a focus on practicing skills of creativity, problem-solving, and collaboration.	▪ 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.	B. Rock, paper, scissors	1. Lead the "Rock, paper, scissors" coding activity. 2. Assess learning with the unit quiz.
			C. Code a board game	Introduce and facilitate the project.
5. 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.	A. Understanding iteration	1. Introduce the concept of iteration. 2. Lead the "Walk a square" unplugged activity.
			B. Coding with loops	1. Introduce three types of loops in Microsoft MakeCode. 2. Lead the three coding activities to demonstrate using 'repeat', 'for', and 'while' loops. 3. Assess learning with the unit quiz.
			C. Get loopy	Introduce and facilitate the project.
6. Mini Project	Provides a review of the concepts covered in the units 1-5,	▪ Code a unique, original program and design and build a physical maker component that uses the	A. Looking back so far, present mini project proposals	1. Provide an overview of the mini-project. 2. Facilitate the review and mini-project plan unplugged activity.

	introduces an independent "mini project," and reinforces the important idea that programming is a process of patient problem-solving.	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	B. Coding and making a mini project	Introduce and facilitate the mini project
			C. Mini project showcase	Introduce and facilitate the showcase activity.
7. 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)	A. Understanding coordinates	1. Introduce the concept of coordinates. 2. Lead the "Battleship" unplugged activity.
			B. Coding animations and patterns	1.Lead the three coding activities to demonstrate using coordinates and LEDs. 2. Assess learning with the unit quiz.
			C. Screensaver or game project	Introduce and facilitate the project.

		individual LEDs and toggle between these two states. ▪ Learn how to check the current on-or-off status of an individual LED and 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.		
8. 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 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 toggle between these two states.	A. Understanding Booleans	1. Introduce the concept of Booleans. 2. Lead the "Two heads are better than one" unplugged activity.
			B. Code a double coin flipper	1. Lead the "Double coin flipper" coding activity. 2. Assess learning with the unit quiz.
			C. Project Boolean	Introduce and facilitate the project.

		▪ Learn how to check the current on-or-off status of an individual LED and 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.		
9. 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.	A. Understanding bits, bytes, binary	1. Introduce the concept of bits, bytes, and the binary number system. 2. Lead the Binary vending machine unplugged activity.
			B. Code a binary calculator	1. Lead the “Binary calculator” coding activity. 2. Assess learning with the unit quiz.
			C. Make a binary cash register	Introduce and facilitate the project.
10. Radio Communication	Introduces the radio functionality of the micro:bit that send and	▪ Understand how to use the Radio blocks to send and receive data between micro:bits.	A. Understanding radio communication	1. Introduce the concept of radio communication between micro:bits. 2. Lead the “Hot or cold” unplugged activity.

	receive numeric and string data between micro:bits and the concept of pair programming.	▪ 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.	B. Explore the radio toolbox	1. Lead the two coding activities using the radio communication blocks. 2. Assess learning with the unit quiz.
			C. Make a micro:bit radio	1. Introduce the concept of pair programming. 2. Facilitate the project.
11. 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	A. Understanding arrays	1. Introduce the concept of arrays. 2. Lead the “Different sorts of people” unplugged activity.
			B. Coding with arrays	1. Lead the three coding activities using Arrays blocks. 2. Assess learning with the unit quiz.
			C. Make a micro:bit musical instrument	Introduce and facilitate the project.
12. Accelerometer	Introduces the accelerometer functionality of	▪ Understand how to use the Accelerometer blocks to sense the micro:bit’s	A. Understanding the accelerometer	1. Introduce the functionality of an accelerometer.

	the micro:bit and covers acceleration, velocity, and other ways to measure the movement of an object through three-dimensional space.	position and movement in three-dimensional space. ▪ Understand the x, y, and z axes and measurement of gravitational force. ▪ Apply the above knowledge and skills to design a unique program using the accelerometer.		2. Lead the "Strike a pose" unplugged activity.
			B. Explore the accelerometer tools	1. Lead the coding activity using the accelerometer blocks. 2. Assess learning with the unit quiz.
			C. Make an accelerometer project	1. Introduce the concept of Global Goals and Do Your: Bit. 2. Facilitate the project
13. Final Independent Project	Provides a review of units 7-12, 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,	▪ Code a unique, original program, and design and build a physical maker component 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 ▪ Accelerometer	Looking back	1.Provide an overview of the final project. 2.Facilitate the review and final project plan unplugged activity.
			Coding and making a final project	Introduce and facilitate the final project.
			Final project showcase	Introduce and facilitate the showcase activity.

	making, and learning process.			