CODING AND ROBOTICS
CURRICULUM GUIDE PRIMARY (DRAFT)

CLASS 1 - 4

SEPTEMBER 2021

MINISTRY OF EDUCATION, TECHNOLOGICAL AND VOCATIONAL TRAINING

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

The Ministry of Education, Technological and Vocational Training (METVT) gratefully acknowledges the following individuals for their contributions to the development of the Coding and Robotics curriculum guide:

Ms. Natasha Jones	**Ellerton Primary (Chairperson)**
Mr. Fernando Carter	**Senior Education Officer**
Ms. Trudy Carter	**St. Joseph Primary**
Mr. Marvin Forde	**Caribbean Institute of Meteorology and Hydrology**
Mr. Shawn Hoyte	**Christ Church Girls’**
Mrs. Arlyn Linton-Jones	**Barbados Community College**
Ms. Shanelle Waithe	**Education Officer**

INTRODUCTION

The Robotics Curriculum focuses on “teleoperated” robots (robots operated remotely), but also introduces concepts of “autonomous” robots. Simple mechanical, and electrical concepts will be introduced using the VEX kit, alongside “block-based” programming, and simple sensors. The goal is to apply this knowledge to automated devices, primarily through the use of “static pre-programmed” action sequences created using “block-based” programming tools and simple sensors.

AIMS

OVERVIEW OF CURRICULUM GUIDE

Coding and Robotics is a fun and engaging way for students to build their critical thinking, logical reasoning and problem solving skills. This document arranges the content by class level and organizes it into proposed modules. Please note, these are recommendations only. Any model can be taught independently or blended with other titles to form a unique learning experience for your students. We recommend using the integrated approach, with authentic, real-world examples to allow your students to see the connections between STEAM and their every day interactions with technology.

The curriculum guide seeks to develop a new generation of creative, innovative systems thinkers that can use technology to positively express their ideas.

COMPETENCIES

DIGITAL TECHNOLOGIES

- Discuss private and personal information
- Develop an awareness of how important it is to protect private data
- Recognise different types of digital systems and give specific examples
- Become responsible users of digital technology
- Build digital skills and competencies
- Develop a working knowledge of hardware and software
- Create a range of digital solutions through guided play and integrated learning
- Explore and experiment with the functions of digital systems
- Analyse the different features of STEAM
- Develop a model of a digital systems
- Understand the relationship between coding robot and STEAM
- Discuss the utilization of the STEAM approach.
- Further develop problem solving skills
- Explore the importance of robotics in everyday life and the role they play
- Identify digital solutions to real world problems

- Adopt a culture of being self-directed, life-long learners
- Further develop problem solving skills
- Solve problems, think critically and work collaboratively and creatively

COMPUTER LANGUAGE

- Learn to use computing terms including the use of general or particular applications to computer language
- Explain how the fundamental components of computer language relate to coding and robotics
- Explore basic graphic and visual representation of binary code
- Show how the components of computer language interact
- Analyse and critique the major types of computer language
- Use skills and techniques to solve basic problems (basic algorithms)
- Connect and engage with others to create algorithms
- Develop and design algorithms to solve real world problems
- Develop computational thinking and problem solving skills
- Explore algorithms and code to create their own interactive project
- Describe the process of creating coordinates for specified location
- Question and investigate how coordinates are identified.
- Acquire, interpret and present information regarding coordinates.
- Demonstrate a willingness to cooperate and take part in group work

CODING AND PROGRAMING

- Explore algorithms and code to create their own interactive stories and games
- Apply coding tools, techniques and strategies to solve problems
- Use coding skills and techniques to solve basic problems
- Develop computational thinking and problem solving skills
- Use computational practices of being iterative and incremental when organizing projects
- Learn to do graphic and visual design/representation
- Develop basic error handling skills to solve code related problems
- Employ computational practices of testing and debugging projects
- Making checks for returned values and exceptions
- Demonstrate their ability to communicate respectfully and effectively through appropriate oral, written and visual presentation
- Able to break up a problem space and design solutions as long as it is within the same platform
- Developing systems that can span multiple technologies and platforms
- Apply computational concepts of parallelism by engaging in computational practices of reusing and remixing projects
- Know your way around the interface and effectively use menus
- Analyse and critique code
- Able better to offer alternatives given requirements based on experience
- Work as a team to address and solve errors in a programming environment

ROBOTICS

- Explore the functions of all the major parts of the robot
- Demonstrate an understanding of specific concepts relating to robotics
- Explore the importance of robots in everyday life
- Develop an appreciation for robotics as a new digital technology
- Learn to visually represent design prototypes
- Create functioning prototypes of robots from everyday items
- Design prototypes using Project Based Learning
- Experiment with simple machines
- Gain an understanding of the correlation between simple machines and robots
- Communicate effectively among peers to complete tasks
- Program robots using robotics software
- Design codes using coding language and software
- Demonstrate an appreciation for the care and handling of robotic equipment
- Employ problem solving techniques
- Think critically and work together in groups
- Collaborate to review and carry out tasks
- Describe and recall experiences to solve coding problems
- Develop a level of mastery in robotics

Module 1: Digital Technologies

SPECIFIC OBJECTIVES	CONTENT	TEACHING and LEARNING ACTIVITIES

CLASS 1		
• Identify and recognise the difference between personal and private information.	Personal (name, age) Private (date of birth, address, passwords) Video: https://drive.google.com/file/d/1spEyqOF7Er2YPIKzeahG6EGOMNh0_l9W/view?usp=sharing	List information about themselves. Classify the information stated into two groups (personal and private). Suggested Assessment: https://docs.google.com/presentation/d/1ukGe_elMU_ExnVEdtB_DLBfkSWVHCbTvTb9i6VCGpPM/edit?usp=sharing
• Recognise and give examples of digital systems	Video presentation or walk around the school to identify - calculators, digital watches, air-conditioners, etc	Brainstorm examples of digital systems and identify their functions. Show a digital system in use.
• Represent patterns in data as pictures, symbols and diagrams	Sequencing (getting ready for school, making lemonade, tying shoe laces, setting down simple calculations)	Discuss the steps needed to complete everyday tasks. Suggested activity: Create a puzzle using card to represent a simple mathematical calculation.
• Characterize and use hardware and software components of digital systems	Hardware (tablet, cellphone, VEX 123 Robot) Software (VEXcode 123)	Use a tablet to record a student relaying weekend news. Manipulate a robot’s actions using the coding software.
• Recognise that a digital system follows specific instructions or commands	Scratch 3.0, VEX robots	Use the ‘**step**’ feature to review the instructions given to a robot to perform a movement.
• Experiment with the functions of digital systems	Tablet, laptop	Describe how to save and retrieve a project.
• Construct a model of a real or imaginary digital system.	Videos of robots performing tasks.	Have a guided discussion about specific problems in their class that they may encounter. Discuss how digital systems can address one or more of the problems presented. (A robot to deliver messages around the school).

● State what is meant by the acronym **STEAM**.	Video Clips, illustrations	Discuss the components of the acronym **STEAM.** Play a game of Hangman where students are given context clues to decipher what the acronym stands for. Use video clips and everyday examples to show the relationship.
● Recognize the importance of robotics in everyday life	Video Clips	Discuss robots currently being used to address real world problems. (Robots being used to deliver food during pandemic, self-driving cars)

CLASS 2		
- Explain the difference between personal and private information.	Personal (favourite colour, first name) Private (date of birth, address, passwords)	Describe the best practices necessary to avoid identity theft. Define identity theft. Suggested Assessment: https://forms.gle/wTzWDi6G9KjjhgEk7
- Identify and explore digital systems	VEX 123 Robot, VEX 123 Coder	Experiment with various algorithms and code to complete a task.
- Recognise patterns in data and represent data as pictures, symbols and diagrams	Sequencing (making conkies, life cycle of a butterfly, process writing)	Providing a prompt, have students order the steps needed to complete a specific task.
- Recognise and use hardware and software components of digital systems	Hardware (tablet, cellphone)	Use a tablet to record a student relaying weekend ews.
- Recognise that a digital system follows specific instructions or commands	CS First, Scratch 3.0, VEX robots.	Instruct a sprite or robot to perform a specific movement.
- Experiment with the functions of digital systems	Tablet, laptop, Scratch 3.0, CS First	Download royalty free images from the internet to upload as sprites.
- Construct a model of a real or imaginary digital system for use in role play	Videos of robots performing tasks.	Design a model of a digital system that can address a real world problem. (A robot to deliver messages around the school).
- Explain the relationship between coding, robotics and STEAM	Video Clips	Discuss the components of the acronym STEAM
- Recognize the importance of robotics in everyday life	Video Clips	Discuss robots currently being used to address real world problems. (Robots being used to deliver food during pandemic, self-driving cars)

CLASS 3		
● Explain the effects of not protecting personal and private information.	Personal (favourite colour, first name) Private (date of birth, address, passwords)	Discuss the possible outcomes due to identity theft.
● Recognise and explore digital systems	VEX 123 Robot, VEX Coder, VEX GO	Experiment with various algorithms and code to complete a task.
● Recognise patterns in data and represent data as pictures, symbols and diagrams	Sequencing (doing laundry with both the washer and dryer)	Providing a prompt, have students order the steps needed to complete a specific task.
● Recognise and use hardware and software components of digital systems	Hardware (tablet, cellphone, laptop)	Create a digital story with photos taken of classmates.
● Experiment with the functions of digital systems	Tablet, laptop, Scratch 3.0, CS First	Edit downloaded royalty free images from the internet to upload as sprites.
● Identify of a real world digital system	Videos of robots performing tasks.	Design a model of a real world digital system. (A car or a loader).
● Recognize robotics in everyday life and its importance	Videos	Discuss robotics being employed to carry out important everyday tasks.

CLASS 4		
Explain the impact of identity theft on an individual and society	Personal (favourite colour, first name) Private (date of birth, address, passwords)	Discuss the possible outcomes due to identity theft on the individual and society.
Identify digital system to carry out specific task	VEX 123 Robot, VEX Coder, VEX GO, VEX IQ	Experiment with various programs to carry out stated tasks.
Recognise peculiarities in features of each system	VEX 123 Robot, VEX Coder, VEX GO Videos, VEX 123, VEX GO, VEX IQ	Create a problem that can be solved only by one of the robots.
Recognize robotics in everyday life and its importance	Videos, VEX GO, VEX IQ	Discuss possible solutions to solving a real world problem using one of the stated robots.
Construct a system to model an existing manual process	VEX 123 Robot, VEX Coder, VEX GO Videos, VEX 123, VEX GO, VEX IQ	Create a solution to an existing manual task.
Identify changes that can be made to an existing robot to carry out more than one function.	VEX GO, VEX IQ	Utilise an existing robot to be multifunctional (adding a claw to grab objects)

Module 2: Computer Language

CLASS 1 ● Examine the basic computing terms. - program - script - command - event - key - running a program	Nowadays, we use computers for everything and as such it is important to understand basic computing terms. ➔ Electronic devices ➔ Scratch 3.0 ➔ Unplugged Activities ➔ Matching activity	Utilise the Scratch interface to learn basic computing terms. Using context clues, define the terms provided. Match the terms to their functions. This can be integrated with: ➔ Spelling and Vocabulary - Sentence stripscan be created and dissected allowing students to ‘find’ specific meanings.
● Name the two major types of computer languages.	Presentation Digital Flashcards ➔ low-level languages ★ Machine language ➢ 1s and 0s (binary) ★ Assembly language ➔ high-level languages ★ Scratch ★ Python	Computers have a language that’s all their own. Similar to different languages spoken by people in different parts of the world; if you want to communicate with a computer you need to learn a programming language. ➔ Discuss how a programming language is designed to communicate instructions to a computer. ➔ Explain that the only language computer hardware can understand is binary code. ➔ Create a *Computer Language Bingo* game to review the vocabulary covered.
● Describe how digital information is represented in 1s and 0s.	➔ binary code ➔ zeros, ones ➔ Obstacle course ★ cones, hula hoops, balls, etc **Resources** ● *CS Unplugged* ➔ **Activity 1**: Count the Dots - Binary Numbers	Create a diagram using a keyboard where the student is giving instructions to a computer on the other side. Over on the next side, have a computer. In the middle will be a box, whose job is to convert our (humans) instructions to computer language. (*See diagram in Appendix 2*) Discuss binary code. Explain that everything seen on a computer is stored using just two numbers - 0 and 1. ➔ Create an obstacle course and have students maneuver the course by

		responding to certain commands. When 0 is voiced the student(s) will walk, with a command of 1, the students will jump.
● Define and use major terms associated with algorithms.	Computers operate by following a list of instructions set for them. ➔ order ➔ sequence ➔ algorithm ➔ instructions Printed or digital copies sequenced tasks. ➔ tying shoe laces ➔ making a mango smoothie	Tell students that today they will pretend you are a robot. They must give you instructions (commands) to reach the door. During the activity encourage the use of specific instructions. Write or project onto the whiteboard, the word *algorithm*. Invite discussion about the steps taken to get ready for school that morning. List the steps provided on the whiteboard. Introduce the terms such as **order** and **sequence**. Explain that an *algorithm is a list of steps taken to complete a task*. ➔ Allow the provision of more every day tasks.
● Design an algorithm to solve a simple real world problem.	Analyse the task and set out a process to complete it. ➔ algorithms ➔ instructions ➔ lists ➔ order ➔ Sequencing	Review the definition of the term '*algorithm*'. Using pictures of sequenced events such as the making of a sandwich. Rearrange and have them placed in the correct order. ➔ Brainstorm daily tasks and have students generate algorithms to successfully complete them. ➔ Provide the class with real scenarios and allow the students to create algorithms.
● Use ordered pairs to locate an object.	Electronic devices ➔ ordered pairs	Explain that ordered pairs are used to locate an object's position on a map. They consist of two

	➔ coordinate plane ➔ x-coordinate ➔ y-coordinate ➔ **Worksheet** ★ Diurnal Animals Ordered Pairs https://docs.google.com/presentation/d/1gvKlsekCT9mbe97913vZlRlUJaLWqlvW9xnBII87zag/edit?usp=sharing	digits - one represents the *x-coordinate* and the other the *y-coordinate*. When these values are combined they give us a location of a given point on a map. Create a simple grid like the one in Appendix 2.1. Invites students to use the key to find the ordered pairs on the grid provided. ● Have students describe how to use coordinates to identify a specific location.

CLASS 2 ● Explain the fundamental concepts of computing through written, visual or oral communication. - selection - commandevent - sequence	➔ Digital flashcards ➔ Teacher created presentation ➔ Differentiate activities ★ Written activity ★ Visual activity ★ Oral activity	Discuss the vocabulary associated with computing. Apply the fundamentals of computing to Scratch as they investigate the blocks used to create simple programs. Have students relay their understanding through a written, visual or oral medium.
● Classify the two major types of computer languages.	Electronic devices ➔ low-level languages ★ Machine language ➢ 1s and 0s (binary) ★ Assembly language ➔ high-level languages ★ Scratch ★ Python	Ask students - 'When you are playing your video games, how does the device know what to do?' Have students discuss their answers, offering guidance where necessary. Review the two major types of computer language. Project onto the whiteboard, a list of computer languages and classify them into two groups.
● Explore cause and effect relationships when designing algorithms.	➔ communication ➔ language that's all their own ➔ different languages ➔ programming language ➔ machine language ➔ code ➔ binary code **Unplugged activity** ➔ Obstacle course	Explain that events are actions that a computer recognizes. When events are triggered, instructions are followed. Brainstorm examples of events such as tapping an object on a tablet screen to select it. Go on to project the term *conditional statement* and explain that it tells the computer to run a different action depending on a specific event. Explore real life if-then cause and effect statements. ➔ What happens if the rain is falling and it's lunch time? Use sidewalk chalk to draw five shapes. Assign roles - one 'programmer' and at least two 'computers'. In this activity the 'computer' will give specific commands (events) that will trigger the 'computers' to perform certain actions.

● Use ordered pairs to locate an object.	Electronic devices Worksheets (printed.digital) ➔ ordered pairs ➔ coordinate plane ➔ x-coordinate ➔ y-coordinate	Encourage the use of programming languages when instructions are given. Review how to use ordered pairs to locate an object's position on a map. Have students describe what it consists of. Project a simple grid and invite students to use the key to find the ordered pairs on the grid provided. Identify a specific object on the grid and have use coordinates to identify a specific location. This can be integrated with: ➔ Social Studies for Map reading ➔ Comprehension for Analysis and Predicting Outcomes ➔ Mathematics for Data Handling

CLASS 3 ● Apply the fundamentals of computing to solve real world problems. - iteration - decomposition - Debugging	Electronic devices ➔ Digital flashcards ➔ Teacher created presentation ➔ Differentiate activities	Provide scenarios which will encompass computing terms. Connect the different concepts learned and have students identify the specific computing term relevant to the scenario.
● Compare and contrast the two major computer languages.	➔ low-level languages ★ Machine language ➢ 1s and 0s (binary) ★ Assembly language ➔ high-level languages ★ Scratch ★ Python	Class discussion on the two major types of computer language. Identify the similarities and the differences between the two.
● Work in groups to create algorithms to solve real world problems.	Electronic devices ➔ Google Docs ➔ algorithms ➔ order ➔ sequence	Review what is meant by the term *algorithm*. Draw connections between algorithms and programs. Invite students to rewrite the definition in their own words. Create groups and provide them with problems to be solved using algorithms. Identify a group leader and have them present the algorithm created.
● Explore cause and effect relationships when designing algorithms.	Electronic devices ➔ Coordinate Plane ➔ Event ➔ Conditional statement ★ if-then ➔ cause and effect	Review what is meant by the term *event*. Brainstorm examples of events such pressing start to play a game or going to the ATM to get funds. Review the term *conditional statement*. Explore real life **if-then** cause and effect statements. ➔ When the La Soufriere erupted in 2021 what happened in Barbados? Ask students to give directions to the window using programming language. Project a

● Identify the x and y coordinates of a plotted point.	Electronic devices ➔ Scratch 3.0 ➔ CS First ➔ whiteboard ➔ board markers ➔ computer ➔ projector ➔ coordinate plane ➔ axis ➔ x- coordinate, y-coordinate	(*coordinate plane*) grid onto the board. Review the interface (**x** and **y** representation and negative numbers). Discuss how coordinate points assist them with figuring out which direction to go in and how far one place is from the other. Add a couple pictures to the grid, invite students to identify the coordinates and explain to their peers how they determined that point. This can be integrated with: ➔ Social Studies for Map reading ➔ Comprehension for Analysis and Predicting Outcomes ➔ Mathematics for Data Handling

CLASS 4 ● Design a program to incorporate the fundamental concepts of computing. - iteration - parallelism - decomposition - conditional statements	Electronic devicesIteration ★ Use of loops ➔ Decomposition ➔ Conditional statements ★ if-then ★ if-then-else	Discuss the meaning of the four highlighted terms. Provide a specific project in Scratch for students to design and code using the four concepts. Encourage students to break the project down into manageable chunks (decomposition). Examine the program to identify where iteration may be useful. Use conditional statements to add an interactive component to the project.
● Describe how digital information is represented in 1s and 0s.	➔ binary ➔ base 2 ➔ base 10 ➔ **CS Unplugged Activity** ★ Working with Binary (pg.16)	Explain that computers use binary to represent information. Highlight the fact that it's called binary because only two digits are used. Show students a base 10 to garner ideas as to why it is called a base 10. Elicit through questioning that binary is known as base 2. Ask students to identify how digital information is represented. ➔ Allow students to complete the *Working with Binary* worksheet ➔ Have them describe how digital information is represented in 1s and 0s.
● Organize ideas in writing or storytelling.	Collaborative grouping Electronic devices ➔ Google Docs ➔ Logical thinking ➔ Procedural thinking ➔ Algorithms ➔ Sequencing	Discuss how a program is a sequence of instructions that a computer follows in an order. Divide students into groups and choose a team leader. Share jumbled action songs via Google Docs with the group leaders and have them share it with their members. Invite the students to imagine they were programming a robot. Have them put the instructions in order to program a robot to dance.
● Categorise computer languages based on use.	Electronic devices, Google slides interactive worksheet	This activity can be created as a Google slide

	➔ High Level vs Low Level ➔ High level ★ Uses words ★ Slower processing time ★ Ease of use for the user ➢ Scratch 3.0 ➔ Low level ★ Machine language ★ Assembly language	interactive activity, or as an unplugged activity where worksheets are created and printed to examine the various computer languages based on use. As an integrative activity in Mathematics, Venn diagrams can be used to sort these languages based on use.
● Compare and contrast an algorithm with a computer program.	Electronic devices ➔ algorithm ➔ computer program ➔ instructions ➔ sequence	Write or project the definition of an algorithm on the whiteboard. Invite students to draw a connection between an algorithm and a program. Allow students to engage in think-pair-share to rewrite the definition in their own words. Ask students to state how the definition compares to that of a computer program.
● Explore cause and effect relationships when designing algorithms	Electronic devices ➔ Scratch 3.0 ➔ Events ➔ Conditional statements ★ **if-then-else**	Ask students the following - ➔ When you place ice cubes in the sun, what will happen? ➔ If you break the school rules, how will you be reprimanded? Allow students to provide their own cause and effect statements. Examine the *if-then-else* block in Scratch. Have students relate the if-then-else block to cause and effect statements.
● Design a program to send a morse code message.	Electronic devices ➔ Morse Code alphabet ➔ Scratch 3.0 ➔ *Music blocks* ★ **play note for** block ➔ Broadcast blocks	Discuss what is Morse Code. Examine the Morse Code alphabet. Determine which **play note for** block from the *Music* category will be used for the **dot** and the **dash** respectively. Students and teacher will work together to build out the code for the **Morse Code alphabet**. They will then use the alphabet to send a short message to another

● Use coordinates to draw shapes.	Electronic devices ➔ projector ➔ coordinate plane ➔ x- coordinate, y-coordinate ➔ perpendicular, parallel ➔ Google Docs ➔ Google Drawing	sprite. Review how to write coordinates correctly. Create a coordinate grid in Google Drawings and paste it into Google Docs. Demonstrate how to build a shape by plotting specified coordinates (points) and joining the points together.

Module 3: Coding and Programming

CLASS 1 ● Understand specific concepts relating to coding and programming. - coding - default - script - command - event - user - input - running a program	Electronic devices ➔ Digital flashcards ➔ Presentation ➔ Video clips ➔ Unplugged Activities	Guided discovery, Brainstorming, Questioning, Viewing Create a mind-map using the words coding and programming. Brainstorm ideas related to coding.
● Identify why Scratch is a programming language.	Electronic devices ➔ Computer code ➔ List of instructions (blocks) ➔ Visual representations ➔ Program ➔ Simple Scratch project (Knock, Knock joke, Catching Papaya Game)	Guided inquiry, Brainstorming, Questioning, Viewing, Problem solving Project on the whiteboard, a simple Scratch project and then show the code. Explain that Scratch is a visual programming language used to build games and interactive stories, music and art.
● Explain the block types used in Scratch 3.0 and their usage. - Stage - Sprite - Sprite's List - Scripts Area - Backdrops - Blocks category - Paint Editor	Electronic devices ➔ Scratch 3.0 ➔ Labelled Scratch interface ➔ Digital flashcards used to label the Scratch interface ➔ Sprites and Backdrops ★ Choose a sprite/backdrop ★ Paint a sprite/backdrop ★ Surprise ★ Upload a sprite/backdrop	Guided inquiry, Brainstorming, Questioning, Viewing, Problem solving Explore the Scratch interface. Navigate the Scratch 3.0 interface identifying the various parts and describing their functions. Describe how blocks react to drag and drop. Suggested Formative Assessment: https://www.liveworksheets.com/ft377322gt
● Manipulate sprites to create a project.	➔ Scratch 3.0 ➔ CS First	Guided discovery, Brainstorming, Problem solving, Collaborative grouping, PBL activity

	➔ scratch.mit.edu	Model how to add, select and delete a sprite. Add an additional sprite. Have students make the last one to be added active.
● Explore the structural and functional purpose of the blocks in the categories below. - Motion - Looks - Sound - Events - Control	Electronic devices ➔ Scratch 3.0 ➔ CS First ➔ scratch.mit.edu Teacher created worksheet which can be printed or done electronically	Guided discovery, Brainstorming, Problem solving, Questioning, Discussion Highlight a block category and investigate its specific function. Provide scenarios and have students drag and drop the particular block into the scripts area. Discuss the ability to add input into the white window.
● Experiment with the scripts area	Electronic devices ➔ Scratch 3.0 ➔ CS First ➔ scratch.mit.edu	Guided discovery, Problem solving, Questioning Demonstrate how to add scripts, rearrange blocks within scripts, drag a group of blocks and remove individual blocks or scripts.
● Describe how a program runs.	Electronic devices ➔ Scratch 3.0 ➔ CS First ➔ scratch.mit.edu ➔ Simple Scratch project to project	Guided discovery, Discussion, Problem solving Create the program shown in Appendix 3.1. Discuss how programs begin at the first block and then run the next code block in the script. Given this, change around the code and allow feedback about how the code has changed. Introduce the process of *debugging* and the importance of testing code periodically.
● Describe how to use a sprite's coordinates to locate it on the stage.	Unplugged Activities Electronic devices ➔ Scratch 3.0 ➔ CS First	Guided discovery, Brainstorming, Problem solving, Questioning A sprite is positioned on the stage using a pair of numbers called the *x-y coordinates*. The

	➔ scratch.mit.edu ➔ Xy-grid background ➔ Coordinates ★ left/right ★ up/down ★ x-coordinates (-240 to 240) ★ y-coordinates (-180 to 180) ➔ Position ➔ Unplugged Activities (*This lesson can also be done unplugged providing students with a printed Scavenger Hunt grid with specific items added at various points, simple map of school on a Scratch grid and are asked to state the position of a specific location*)	x-coordinate identifies how far left or right the sprite is across the stage. The y-coordinate shows how far up or down the sprite is. ➔ Add one sprite to the stage. Change the backdrop to the **Xy-grid**. Highlight the **current sprite info** and have students focus on the x and y coordinates. Identify the sprite's current location and then move it to specific points, describing what happens to x and y values. ➔ With the Xy-grid background active, create a simple program to position the sprite using the '**go to x: y:**' block. ➔ Add additional sprites on the stage and have students identify their position.
● Code arrow keys to move a sprite	Electronic devices ➔ Scratch 3.0 ➔ CS First ➔ sratch.mit.edu ➔ Motion blocks ★ Move blocks ➔ '**when key pressed**' block left arrow, right arrow, up arrow, down arrow	Guided discovery, Brainstorming, Problem solving, Collaborative grouping, PBL activity Explain that when creating a program or building code it is always good to have a plan. Discuss the desired outcome and design a plan to execute the process. Construct four simple scripts to control a sprite using the arrow keys. Use the '**when space key pressed**' hat block and add a '**move 10 steps**' block. Allow manipulation of the number of steps.
● Manipulate a sprite's orientation on the stage	Electronic devices ➔ Scratch 3.0 ➔ CS First ➔ scratch.mit.edu ➔ Direction Terms ★ All around	Guided discovery, Brainstorming, Problem solving Add blocks to code the sprite to move back and forth across the stage. Encourage discussion about what is happening. Bring attention to the

	★ Left/Right ★ Do not rotate ➔ Motion block ★ **‘if on edge bounce’**	**current sprite info** and highlight the **Direction** window. Identify the three *rotation styles*. Test the code after each orientation change and describe the effect.
● Manipulate Repeat and Forever loops	Electronic devices ➔ Loops ★ Repeat A repeat loop repeats the blocks inside it for a specified number of times and then runs the next blocks in the script. ★ Forever A forever loop repeats the block(s) inside it forever.	Guided discovery, Brainstorming, Problem solving, Discussion Run the code in Appendix 3.2 in full screen. Invite discussion about the possible code used. After the code is shown, ask which block controls the sprite. Using context clues, guide the discussion towards what is a **loop**. Explain that the orange **Control** blocks control the actions of a sprite. Ask which block can be used to allow for a finite movement.
● Design and create a simple dialogue between two sprites.	Electronic devices ➔ **Event blocks** ★ *when green flag clicked* ★ *when this sprite clicked* ★ *when key space pressed* ➔ **say hello for 2 secs** block ➔ **wait** block	Guided discovery, Brainstorming, Problem solving, Collaborative grouping, PBL activity Use the **wait** block to control the timing of the conversation. ➔ Add two sprites ➔ Make the sprites face each other ➔ Add a backdrop Discuss how conversations flow. ➔ Code the first sprite to say something ➔ Code the second sprite to wait and then respond

CLASS 2 ● Animate sprites.	Electronic devices Scratch 3.0 ★ Looks blocks ➔ CS First ➔ Scratch.mit.edu ➔ Animal Antics ➔ https://www.stemdetectivelab.com/scratch-coding-camp/animal-antics-badge/	(*Suggested activity - Animate the letters of their name, initials or a specific word*) Guided discovery, Brainstorming, Collaborative grouping, PBL activity Add an interesting sprite. Explore the Looks blocks to animate the sprite. ➔ change size by block ➔ set size to block ➔ change color effect by 25 ➔ set color effect to 0 ➔ clear graphic effects ➔ show ➔ hide
● Create an animated dance.	Electronic devices ➔ Scratch 3.0 ➔ Looks blocks ★ **next costume** ➔ Control blocks ★ **repeat** block ★ **forever** block ➔ Motion blocks	Guided discovery, Brainstorming, Collaborative grouping, PBL activity Animate a Sprite as it moves using more than one costume. ➔ Add the Ballerina sprite ➔ Change the **rotation style** of the Sprite. ➔ Create a simple program to make the sprite appear as though it is dancing. (*See Appendix 3.3*)
● Create a digital story ➔ The Role of Community Workers ➔ A story written in Composition ➔ Steps needed to complete a calculation	Electronic devices ➔ Scratch 3.0 ➔ CS First **Resources** Costume Animation ➔ https://csfirst.withgoogle.com/c/cs-first/en/storytelling/dialogue/extensions/costume-animation-p	Guided discovery, Brainstorming, Collaborative grouping, PBL activity Program a conversation between two characters to explore the role of dialogue in storytelling. Use a story starter or a title to plan a short story. ➔ Choose characters ➔ Add a scene (backdrop)

	art-1-of-2.html	➔ Add dialogue ➔ Bonus - animate sprites to bring the story to life Suggested Rubric: https://docs.google.com/document/d/1DNlmxIStvU9RqSct58LUv_vi-mm6M-6bmEYA51IJoMg/edit?usp=sharing
● Add scenery and music to Scratch projects.	Electronic devices ➔ Scratch 3.0 ➔ CS First ➔ scratch.mit.edu	Guided discovery, Brainstorming, Collaborative grouping, PBL activity Improve on the code created in previous projects by adding sound and additional backdrops. ➔ Select a sprite ➔ Click on the Sounds tab ➔ Choose a sound from the library ➔ Return to the Code ➔ Add the sound to the sprite’s script
● Investigate the directionality of sprites	Electronic devices ➔ Scratch 3.0 ➔ computers use numbers (degrees) to indicate an exact location ➔ In Scratch the direction numbers are between -180 and 180 ➔ compass ➔ negative and positive numbers ➔ point in direction block	Guided discovery, Brainstorming, Collaborative grouping, PBL activity Guided discussion about how humans follow directions compared to that of computers. Use a compass to tap prior knowledge and then introduce the four main directional values using Scratch’s compass - ➔ 0º- facing up ➔ 90º - facing right ➔ -90º - facing left ➔ -180º and 180º- facing down This lesson can be integrated with Social Studies (cardinal points).

● Demonstrate how to control the timing of events.	Electronic devices ➔ Scratch 3.0 ➔ CS First ➔ Characterisation ★ Describe a character in a story and explain how his/her actions contributed to the sequence of events. ➔ **wait** block Resource: https://docs.google.com/presentation/d/1H1n5rQODJMC4u53TnnhAPZlhECqb9wK16-m0zj_hl-A/edit?usp=sharing	Guided discovery, Brainstorming, Collaborative grouping, PBL activity Add a sprite to the project to investigate the effects of changing the directional value. Provide details about a character who is missing an assignment and use code to describe the character's actions, thoughts and words. ➔ Use Event blocks to trigger a series of code ➔ Sequence at least four say blocks ➔ Use motion blocks to personalise the story ➔ Use wait blocks to control the timing
● Describe the process of broadcasting and create a "knock knock" joke.	Electronic devices ➔ Scratch 3.0 ➔ CS First ➔ *Events* blocks ★ **broadcast** block ★ **when I receive broadcast message** block	Guided discovery, Brainstorming, Collaborative grouping, PBL activity Plan a simple knock knock joke. Discuss the dynamics and create a code to represent the joke. Use the **broadcast** blocks to communicate between sprites.
● Use code to tell fun and interactive stories ➔ Digital Show and Tell ★ People in the Neighbourhood ★ Diurnal and Nocturnal Animals ★ Even and Odd Numbers ★ Nouns and Adjectives	Electronic devices ➔ Scratch 3.0 ➔ CS First ➔ *Looks* blocks ★ **switch costume to** block ★ **next costume** block ★ **switch backdrop to** block ★ **next backdrop** block ➔ *Events* blocks ★ **when backdrop switches to** block	Guided discovery, Brainstorming, Collaborative grouping, PBL activity Design a simple story or info. Make the project interactive by using for example **when this sprite clicked** block. ➔ Add music to the story ➔ Switch backdrops ➔ Add a third sprite ➔ Change costumes

● Narrate a story in Scratch.	Electronic devices Collaborative activity - collaborate in **Google Docs** to plan the story ➔ Scratch 3.0 ➔ CS First ➔ *Events* blocks ➔ *Motion* blocks ➔ *Control* blocks ➔ *Sound* blocks Suggested Rubric: https://docs.google.com/document/d/1DNlmxIStvU9RqSct58LUv_vi-mm6M-6bmEYA51IJoMg/edit?usp=sharing	Guided discovery, Brainstorming, Collaborative grouping, PBL activity Create a project that tells the same story from two points of view. ➔ Sequences **say** blocks to tell a story in 1st person or third person. ➔ Use *Events* blocks to trigger a series of code. ➔ Use **wait** or **broadcast** blocks to control timing ➔ Use *Motion* blocks, sound effects and animation to personalise the story
● Use the Paint Editor to create new costumes for existing sprites	Electronic devices ➔ Scratch 3.0 ➔ Costumes tab ★ *Paint Editor* ➢ *Select tool* ➢ *Reshape tool* ➢ *Fill tool* ➢ *Brush tool* ➢ *Eraser tool* ➢ *Text tool* ➢ *Line tool* ➢ *Circle tool* ➢ *Rectangle tool*	Guided discovery, Brainstorming, Collaborative grouping Select the sprite to be modified. Open the Paint Editor by clicking on the **Costumes** tab. Use the tools provided to create new costumes for a sprite. Animate the sprite using its new costumes.
● Design and animate a new sprite	Electronic devices ➔ Craft supplies (paper, card, scissors, crayons) ➔ Scratch 3.0	Guided discovery, Brainstorming, Collaborative grouping, PBL activity Design a character they would like added to the *Sprite's Library* and make a model. Use the *Paint*

	➔ Costumes tab ★ *Paint Editor* ➔ Decompose problems into smaller, manageable pieces to control project succession.	*Editor* to create and name their sprite. Animate the new sprite. This activity can be integrated with ➔ Visual Arts - the student can design and build his/her sprite using cardboard or other media. ➔ Storytelling - the student could speak about the character providing additional details.
● Name the major components of a good game.	Electronic devices ➔ Game design ★ *Variables* - score, timers ★ Sound effects ★ Sprite iterations ★ Levels ➔ Scratch 3.0	Guided inquiry, Brainstorming Ask, ‘What do you think makes a good game’? Record the responses and Explain that a good game tests your skills. It can include movement, speed and obstacles. Review how using the concepts taught previous to design and program a simple game.
● Design and code a simple game. ➔ Hungry Green Monkey	Electronic devices ➔ Game design ➔ Scratch 3.0 ➔ Decompose problems into smaller, manageable pieces to control project succession. ➔ *Sensing* blocks ★ **touching** block ➔ *Control* blocks ★ **if-then** blocks An **if-then** block wraps around other blocks and uses a true or false question to control whether those blocks are runned or skipped. If the answer to the question is true, the blocks inside will run.	Guided discovery, Brainstorming, Collaborative grouping, PBL activity Using skills and concepts mastered, design and code a game. ➔ Code a game to make a sprite move around the stage ‘eating’ bananas. ➔ The player controls the sprite by pointing to a place on the screen (stage) using the cursor (mouse). ➔ The sprite will then move towards the banana. ➔ If the cat touches the banana, the banana disappears as if it’s eaten.

CLASS 3 ● Use code to tell fun and interactive stories ➔ Verb Simile ➔ Metaphor	Electronic devices ➔ Scratch 3.0 ➔ CS First ➔ *Events* blocks ➔ *Motion* blocks (*used to show what the figurative language seems to mean versus what it actually means*) ➔ *Control* blocks ➔ *Sound* blocks **Resources** ➔ https://csfirst.withgoogle.com/c/cs-first/en/figurative-language/figurative-language/explore-your-figurative-language.html	Guided discovery, Brainstorming, Collaborative grouping, PBL activity Create a simple Scratch project using the concept of broadcasting. ➔ Add figurative language using **say** blocks ➔ Use *Motion* blocks to move the characters across the stage ➔ Add at least three object sprites to help describe the figurative language ➔ Add backdrops to help show meaning ➔ *Broadcast* blocks are used to trigger code
● Narrate a story in Scratch. ➔ Sports Day ➔ AgroFest ➔ A Fictional Story	Collaborative activity - collaborate in **Google Docs** to plan the story ➔ Scratch 3.0 ➔ CS First ➔ *Events* blocks ➔ *Motion* blocks ➔ *Control* blocks ➔ *Sound* blocks	Guided discovery, Brainstorming, Collaborative grouping, PBL activity Create a project that tells the same story from two points of view. ➔ Sequences **say** blocks to tell a story in 1st person or third person. ➔ Use *Events* blocks to trigger a series of code. ➔ Use **wait** or **broadcast** blocks to control timing ➔ Use *Motion* blocks, sound effects and animation to personalise the story
● Create an interactive story with code.	Two characters meet in a world and discover a surprising object. Students get to decide what happens next by creating a story with code. Electronic devices	Guided discovery, Brainstorming, Collaborative grouping, PBL activity Design and build a story that's interactive. Use blocks such as **when this sprite clicked** or **ask**

	➔ Scratch 3.0 ➔ CS First	found in the *Events* and *Sensing* categories respectively. ➔ Choose the setting. ➔ Choose sprites and add code for a dialogue. ➔ Animate the objects ★ Grow/shrink ★ Colour changes ★ Make an object bounce ➔ Add mystery with a pulsing object ➔ Add sound This lesson can be integrated with all of the core subjects. ➔ Mathematics - How to add fractions ➔ General Science - Comparing Vertebrates ➔ Social Studies - Transportation ➔ Language Arts - What is an Adverb
● Add a title to a project.	Electronic devices ➔ Scratch 3.0 ➔ CS First ➔ Paint Editor ★ *Text* tool ★ *Select* tool ★ *Fill* tool ★ *Rectangle* tool ➔ https://csfirst.withgoogle.com/c/cs-first/en/an-unusual-discovery/an-unusual-discovery/extensions/animate-a-story-title.html	Guided discovery, Brainstorming, Collaborative grouping, PBL activity Brainstorm the title of a project they would like to program. Click ‘**Choose a Sprite**’ and select ‘**Paint**’. Click the ‘*Text*’ tool and type the title. Use the *Select* tool to highlight and resize. Change the font using the *Text* tool and the colour using the *Fill* tool. Emphasis the title by adding a box around the border. Use the backward tool to show the text in front of the box. Animate the title.
● Animate an object to make it bounce.	Electronic devices ➔ Scratch 3.0 ➔ CS First	Guided discovery, Brainstorming, Collaborative grouping, PBL activity

	➔ *Motion* blocks ★ **Glide 1 secs to random position** block ★ **Go to x: y:**block ➔ *Control* blocks ★ **repeat** block	Explore the **glide** Motion blocks as well as the **go to x: y:** block. Manipulate these blocks to make the sprite appear as though it's bouncing on the stage. ➔ Select the object sprite ➔ Add a *glide 1 secs to random position* block; *repea*t block ➔ Add the *go to x: y:* block to make the sprite move to a specific position when it stops bouncing.
● Create a two player racing game.	Electronic devices ➔ Scratch 3.0 ➔ CS First ➔ Control movement with the keyboard ➔ *Events* blocks ★ **when key space pressed** block ➔ *Motion* blocks ★ **move 10 steps** block ★ **turn left 15 degrees** block ★ **turn right 15 degrees** block ➔ *Control* blocks ★ **repeat until** block	Guided discovery, Brainstorming, Collaborative grouping, PBL activity Review how to code the up, left and right arrow keys to control the movement of the car sprite. Discuss how to make the sprite move smoothly with the **repeat until** block. Code the player two to move using different keys.
● Describe how to make and use Scratch variables	A variable is like a labelled box in which you can store data such as **words** or **numbers**. The data stored in a variable is called its **value**. Electronic devices ➔ *Variables* blocks ★ **Make a Variable** ➢ https://www.youtube.com/watch?v=wS1QETPfgGE	Guided inquiry, Discussion, Brainstorming Discuss score and timers in games. Explain that variables are used to control time and keep score. Select the *Variables* block category, click **Make a Variable**. Add the variable name to be used.

● Create a new block. ➔ say a random greeting in any language ➔ draws a particular shape ➔ states specific vocabulary ➔ states Barbadian facts	*My Blocks* offer functionality. Scratch allows you to create your own blocks and use them just like the built in blocks, i.e. *creating your own functions*. Electronic devices Resources ➔ https://en.scratch-wiki.info/wiki/My_Blocks	Guided discovery, Brainstorming Create the program needed with the necessary blocks. ➔ Select the *My Blocks* ➔ Then click '*Make a Block*' ➔ In the empty block, type in the name of the desired new block. An event style block appears in the **Scripts Area**. Once this new block has been defined we are able to use it like any other block.
● Design and code a game to catch items falling from the sky. ➔ Catch a Breadfruit Game ➔ Catch Prime Numbers ➔ Catch Specific Flags	Electronic devices ➔ Game design ★ Make copies of sprites ★ Sprite animation ★ Time delays ★ Variables ➔ Scratch 3.0 ➔ CS First ➔ *Control* blocks ★ **if-then-else** block ➔ Decompose problems into smaller, manageable pieces to control project succession. ➔ Add Extension ★ **Text to Speech** ➔ Give specific instructions	Guided discovery, Brainstorming, Collaborative grouping, PBL activity Using skills and concepts mastered, design and code a game which catches items falling from the sky. Utilise variables to make the game more interesting. Incorporate the *Text to Speech* feature in game. ➔ Choose Add Extension from the bottom left of the *Code* section ➔ Select Text to Speech

CLASS 4		
● Use the *Join* block appropriately.	Electronic devices ➔ Create variables to hold the data.Use the **join** block to link values together ➔ Add the **ask** block	Guided discovery, Brainstorming, Collaborative grouping, PBL activity ➔ Create a simple code using the join block to make a sprite greet you with your name. (*See Appendix 3.4*) ➔ Design a times tables quiz where ten questions about the 1 to 12 times tables are asked. Allow for input of answers. ➔ score for each correct answer. (*See Appendix 3.5*)
● Create drawings with iteration	Use *variables* in repeats and iterative development ➔ **Pen** blocks (Add Extension) ➔ Add **variables** ➔ Use **Loops** ➔ Use **broadcast** blocks ➔ Scratch 3.0	Guided discovery, Brainstorming, Collaborative grouping, PBL activity Updating **variables** in repeats and iterative development. Create two balls in the *Paint Editor* ➔ Code the balls to bounce around the stage drawing and changing patterns.
● Explore the concept of *Loops*, *Events* and *Conditionals*.	Electronic devices ➔ CS First ➔ Play notes on musical instruments (*See Appendix 3.3*) The *repeat until* loop block keeps looping until a specific action happens. The *wait until* block completes the command specified in its input window before it moves onto the next command.	Guided discovery, Brainstorming, Discussion Investigate the two conditional blocks from the *Control* blocks category. Compare the **repeat until** block with the **wait until** block. Create a project that uses the concepts learnt.

● Compute simple calculations in Scratch	Four blocks in the green **Operators** section can do math with whatever values inputted. ➔ Arithmetic Operators ★ Add ★ Divide ★ Subtract ★ Multiply ➔ *Looks* blocks ★ **say** block	Guided discovery, Brainstorming, Discussion, Collaborative grouping, PBL activity Investigate the use of *Operator* blocks. ➔ Put an Arithmetic operator inside of a **say** block to compute a calculation. Describe what happens when the script is triggered. ➔ Experiment with more than oneArithmetic Operator to compute more difficult calculations.
● Design code to ask questions and receive responses. - Review exercises - Exit tickets - Short quizzes - Survey	Ask the audience a question and make something happen based on the response. ➔ Scratch 3.0 ➔ CS First ➔ *Sensing* blocks ★ **ask** block ★ **answer** block ➔ *Control* blocks ★ **if-then-else** block ➔ *Operator* blocks ★ **equals operator (**comparison operator**)**	Guided discovery, Brainstorming, Collaborative grouping, PBL activity Decide the content and then brainstorm questions to be asked and what is to happen based on the responses received. Make the project interactive and get the audience involved in the activity. ➔ Add an *ask* block to ask the audience a question ➔ Check the user's answer by using the *if-then-else* block; the *equals operator* and the *answer* block ➔ Add s*ay for 2 secs* blocks to program different things to happen based on the user's response Sound and graphic effects can also be added.

Module 4: Robotics

CLASS 1 ● Identify parts of a robot: - Controller - Body - Mobility - Power - Sensors - Tools	Video clip, pictures, assembled robot (VEX Kit), http://prime.jsc.nasa.gov/ROV/types.html	Guided Discovery,Viewing, Brainstorming, Discussion From the videos and the display of VEX robot, discuss and brainstorm to identify various parts of a robot: - Controller - Body - Mobility - Power Sensors - Tools
● State the function each part of the robot	Video clips, pictures of robot, assembled robot (VEX Kit)	Viewing, Guided Discovery, Brainstorming, Discussion Unplugged Activities - 1. Singing/Moving Game: Head, Shoulders, Knees and Toes 2. Body Awareness Exercise - How do human joints move?
● Design basic prototypes of robot arm	Video clips, Paper, bristol board, cardboard, pencils, crayons, coloured markers, magnets, paperclips www.tryengineering.org www.mos.org/robot/robot.html www.usfirst.org	PBL activity, Brainstorm ideas, Think-Pair-Share, Discussion on how robots are used in everyday life. Discuss why robots and humans move in different ways? **Unplugged Activity** - *Scavenger Hunt* (collect as many items (paperclips) using a small magnet) Groups design a prototype cardboard arm to be used in the medical field in transporting medical equipment - Designs can be used to identify various modes of transportation. Share ideas on designs. Engage in discussion to garner ideas on various designs for specific

		prototypes.
● Create a basic robot using everyday items	Paperclips, pencils, plastic straws, ping pong balls, styrofoam/plastic cups,	Guided Discovery, Discussion, Collaborative Grouping, PBL activity **Unplugged Activity** - Construct a prototype of a cardboard arm using a selection of everyday items. Each team must agree on a design for the robot arm and identify what materials will be used.
● Operate a prototype to complete a specific task or function	Robots made from everyday items, Questionnaire, worksheet	Collaborative Grouping, PBL activity, Peer Assessment, Discussion, Viewing, Brainstorming, Guided Discovery, Questioning **Unplugged Activity** - The robot arm must be at least five inches in length, and be able to pick up an empty styrofoam/plastic cup, and hold for at least five seconds, before the cup is released. Team should showcase their design before testing the prototype. Through discussion, a definition should be identified to describe simple machines as tools used to make work easier.
● Define a simple machine	Simple Machine -	**Unplugged Activity** - Egg/Lime and Spoon Relay Race (Each team member will be given a spoon/plate, they must pass the lime/egg to

	A simple machine is a tool that is used to make work easier. Video clips, exhibits of simple machines, limes, plastic spoons, small plastic plates	another team member without dropping the lime while travelling a distance. Each team member must travel with the egg/lime.)
● Identify simple machines - Inclined plane - Lever - Screw - Wedge - Pulley - Wheel and Axle	VEX IQ Curriculum handout	From the handout, all of the simple machines should be identified. Engage students in discussion as to how each machine is used in everyday life - give examples of simple machines in action. - Inclined plane - Lever - Screw - Wedge - Pulley - Wheel and Axle
● Create at least one of the simple machines	Cardboard, string, straws, cotton reels, bristol board	Collaborative Grouping, PBL activity Groups will be asked to select one of the simple machines introduced, and use the items given to build a functioning prototype.
● Problem solve to fix/alleviate problems that may arise	VEX Curriculum, STEM Labs	Discussion, Problem Solving, Collaborative Grouping, PBL activity

● Explain how friction works as it relates to robotics - Friction Friction is the force that resists motion through the rubbing of one object against another.	Resource: http://www.stemmom.org/2012/09/friction-inquiry-lab.html PDF: https://docs.google.com/open/?id=0B5vZUViZFTxaNXAzcliGRTRiSG8 Video clips: http://www.youtube.com/watch?v=miGZR4szu8 https://www.youtube.com/watch?v=CTLXubXOTUQ http://www.youtube.com/watch?v=VUfqiSeeZng VEX IQ Curriculum handout, rope	Discussion, Collaborative Grouping, PBL activity
● Build a basic robot using designated kits	VEX Kit, VEX Education (copy of robot assembly) worksheet, electronic devices Resource: https://www.youtube.com/watch?v=bGTPDp2cNtM&list=PLvvcc7S26YEgp60=fNJwh64aj9ywiZ79Ta&index=5	Discussion, Guided Discovery, Teamwork, PBL activity, Experimentation **Unplugged Activity** - *Tug of War* - Students will be asked to give a hypothesis as to what they think will happen if they performed Tug-of War with shoes on a variety of surfaces. Split class into two teams. Teams will use a rope to play tug of war on three different surfaces - grass, smooth and rocky. Then discuss if their predictions were accurate.
● Operate basic build robot using teleoperation	Obstacle course, electronic devices, VEX kits	Guided Discovery, Collaborative Grouping, Teamwork, PBL activity **Introduction to ModKit**

● Navigate and control using teleoperation through an obstacle course	Obstacle course, electronic devices, completed basic build of the robot	Each team will be required to construct the basic build of the robot using the instructions provided in the kits. This activity will help to tie in the parts of the robots introduced earlier. Basic build should include - brain, motor, wheels and gears. Charge - controller Connect joystick to brain Connect brain to computer Checking battery voltage Turn on motor Connect to ports to brain Download program Collaborative grouping, PBL activity, Guided Discovery Groups will be given a copy of the instructions to complete the basic build. Groups should designate duties for each member so that the basic build can be completed effectively **Unplugged Activity** - *Visual Arts* - An obstacle course can be designed and constructed by the students as part of a Visual Arts lesson. The obstacle course should be designed to test the concept friction previously covered. Collaborative grouping, PBL activity At this stage, groups should have completed the basic build, and be able to manoeuvre the robot

		around an obstacle course remotely. **Activity** - After the build is complete, to test the build of the robot, teams can engage in timed and dexterity events in and around the classroom.

CLASS 2 ● Determine the various functions of all the parts of a robot: - Controller - Body - Mobility - Power - Sensors - tools	Display of unassembled VEX kit parts Resource: http://prime.jsc.nasa.gov/ROV/parts.html	Discussion, Guided Discovery, Questioning, Collaborative grouping Groups will be given a kit. From the handout students should be able to examine and explain the basic function of each part of the robot selected.
● Classify a robot by the way it functions: - Autonomous - performs behaviors or tasks with high degree of freedom - Teleoperation - the control of a device or machine remotely	Completed basic build of VEX robot, cardboard, VEX coloured blocks, pencils, sheets of paper Video clip: https://www.youtube.com/watch?v=CDDGBcs0TFM https://www.youtube.com/watch?v=IFXwa2NOkqs https://www.youtube.com/watch?v=aaTziqnWapk	Guided Discussion, Collaborative grouping, Teamwork, PBL activity **Unplugged Activity** - A maze will be set up on the floor. Each group will be instructed to navigate the maze to collect a series of blocks. Groups will be asked to pretend that one of their members is a robot. That member will be blindfolded and the other members are to assist that member to navigate through the maze successfully to the end (teleoperation). After this, each group will write all of the instructions to navigate the maze on a piece of paper. They will give the instructions to a selected member of the group. This member will read and follow the instructions to collect the blocks and navigate through to the end of the maze (autonomous).
● Explain the various types of robots	Resource:	Discussion, Questioning

- Cartesian robot - Cylindrical robot - Spherical robot - SCARA - Articulated robot - Parallel robot	http://prime.jsc.nasa.gov/ROV/types.html www.tryengineering.com Video Clips: https://www.youtube.com/watch?v=HX6M4QunVmA https://www.youtube.com/watch?v=6L-V4xzUcmM	Engage students in a discussion as to the use of the various types of robots and the various industries that they can be found.
● Explain how centre of gravity works as it relates to robotics - Centre of gravity/Mass - is the place in a system or body (such as a robot) where the weight is evenly distributed and all sides are in balance.	Broomsticks Resource: https://www.youtube.com/watch?v=CZsZz24QwNM https://www.youtube.com/watch?v=Fw6Ra_2oZM4 https://www.youtube.com/watch?v=2WUdHBso3VK	Discussion, Questioning **Unplugged Activity 1**: Limbo - A game of limbo requires two children to hold each end of a broomstick. Encourage the other children to take turns trying to go under the limbo stick while bending backward. Let the children take turns holding the stick lower and lower. Discuss at which point they fall backward and why they fell.
● Build and use simple machines to see how the work - Plane - Wedge - Screw	VEX Kits, VEX Curriculum worksheet	Guided Discovery, Discussion, Collaborative Grouping, PBL activity Teams will use objects from the kit to construct their version of the three simple machines - plane, wedge and screw.
● Construct a medical robotic arm using everyday items	Ping pong balls, small plastic plates, rubber bands, clothes pins, paperclips, plastic cups	Collaborative Grouping, PBL activity, Peer assessment

	Resource: Hydraulic Powered Robotic Arm https://www.youtube.com/watch?v=P2r9U4wkjcc	**Unplugged Activity** - *Smooth Operator* - Each team of engineers have been given a challenge to design a surgical instrument. To test the instrument four small objects must be removed from a small plate without knocking over any of the other objects on the plate. All components of the instrument must be physically connected. All objects must remain intact for additional study by the medical team, Visual Arts - Each team should draw a sketch of the agreed upon design prior to construction. All materials should be outlined, and the number of parts that are needed.
- Build simple machines using designated kit	Designated VEX kits, VEX Curriculum worksheet (simple machines)	Collaborative Grouping, Guided Discovery, PBL activity Teams should select the simple machine they will build. Each team will test the functionality of their machine.
- Build a basic robot attaching a functioning arm	Designated VEX Kits, VEX Curriculum worksheets (Clawbot)	Collaborative Grouping, PBL activity Each team will be given an unassembled kit. Each team will select a team leader, the leader will disseminate specific tasks to each team member. Teams will be instructed that they have a specified time in which to complete their build. Team member 1 - Sorting kit pieces 2 - Assembly of base of robot 3 - Assembly of clawbot 3 - Assembly of drivetrain

● Navigate a robot’s teleoperation through an obstacle course moving turning (90, 180, 360 degrees) and collecting blocks	Designated VEX Kit, Obstacle course, VEX Coloured Blocks	Collaborative Grouping, PBL activity After the assembly of the clawbot. Each team will be instructed to navigate the clawbot tele-operated through an obstacle course ensuring that the robot can perform various turns (90, 180, 360 degrees) while collecting the coloured blocks and returning back to base.
● Design a simple autonomous program	VEX programming software, Clawbot build, electronic devices	Collaborative Grouping, Teamwork, Guided Discovery, PBL activity Each team will be instructed to design a simple program that will allow the clawbot to successfully navigate the obstacle course. The program design would entail the robot moving a specified distance to collect two or more coloured blocks, then turn and return to base, This activity can be an integrative lesson in Mathematics focusing on angles and measurement.
● Program a robot to navigate and control a robot with arm appendage (clawbot) to move and carry objects	VEX programming software, Clawbot, Obstacle course, electronic devices	Collaborative Grouping, PBL activity Each team will test their autonomous program to determine whether the clawbot can successfully navigate the obstacle course.

CLASS 3 ● Demonstrate an understanding of specific concepts related to robotics - Power - is the rate at which work is done. - Torque - is a force directed in a circle, most often rotating an object. Torque is a spinning force. - Speed - is a measure of how fast an object is moving.	VEX curriculum handout Resource: https://www.youtube.com/watch?v=zGnp52kZ9gE https://www.youtube.com/watch?v=4PIhvPTONug https://www.youtube.com/watch?v=eUeGNe4eDAw&t=12s	Discussion, Viewing, Guided Discovery, PBL activity **Unplugged Activity** *Torque* - Using the door of the classroom, demonstrate how difficult it is to open if you push right next to the hinges. Explain that in this case, the object you are trying to rotate is the door. The pivot point/axis of rotation is at the hinges, and thus, in order to maximize the lever arm, the doorknobs where we normally push are located as far as possible from the axis of rotation. This is done to maximize the amount of force needed to open the door (also known as rotating the body about an axis). This can be an integrative activity with **Mathematics** - introducing the use of tables and graphs. **General Science** - introduction of what is a force **Unplugged Activity** (PBL) - Basketball Robot - Design and build a ‘robot’ basketball player that can shoot a ping pong ball accurately into a ‘net’ one foot above the floor or desk and six feet from the robot. The final design will be allowed ten attempts at the ‘net.’ Each ball that does not make it into the net will have the position they ‘land’ marked with masking tape, and the distance from the ‘net’ will be recorded in a table chart. This is an integrative activity with Mathematics for data representation and measurement.

● Complete an install of latest version of firmware to the brain of the robot	Electronic devices, VEX Kits Resource: https://www.youtube.com/watch?v=ELo481zsbTI&list=PLvce5HOtetEt6H8RZcbfeKTbKfQ-XGExN https://www.youtube.com/watch?v=tU5E2QIsHIo&list=PLvce5HOtetEt6H8RZcbfeKTbKfQ-XGExN&index=2 https://www.youtube.com/watch?v=bGTPDp2cNtM&list=PLvce5HOtetEt6H8RZcbfeKTbKfQ-XGExN&index=4 https://www.youtube.com/watch?v=hR1968xskSQ&list=PLvce5HOtetEt6H8RZcbfeKTbKfQ-XGExN&index=5	Collaborative Grouping, Viewing, PBL activity From the videos provided, along with the VEX Curriculum. teams will be asked to work in their groups identifying the various sensors found in the kits.This activity will identify that the robot has approximately five sensors: - Gyro - Bumper - LED Light - Distance - Colour
● Understand the use of a sensor Sensor - a robot using sensors to gather information about their environment so that they can react to their surroundings	VEX Kits, electronic devices, obstacle courses	Collaborative Grouping, PBL activity Teams will build the basic build of the robot and attach each type of sensor to understand its functionality. **Activity** - As a Visual Arts integrated activity, each team can design a simple obstacle course to help test their robot. This lesson can be done using the robot teleoperation, and for a challenge teams can design a simple automated program to test each of the sensors individually.

● Build a simple robot adding sensors - Bumper - Gyro	VEX kits, Basic build robot Resource: https://www.youtube.com/watch?v=wMT8MBSB-T8 https://www.youtube.com/watch?v=ELo481zsbTI	Collaborative Grouping, Teamwork, PBL activity From the basic build, teams should be able to further assemble the robot attaching the following sensors: - Bumper - Gyro
● Construct a more complex build using sensors from the specified kit - Bumper - Gyro	VEX kits, electronic devices, clawbot, obstacle course	Collaborative Grouping, Teamwork, PBL activity Each team will design an automated program that will require the robot to utilize the attached sensors. The robot should be able to move forward a specified distance of the obstacle course. Code robot to include a "when" condition code so that when the bumper sensor is touched it turns and moves in another direction. In other words, Start, forward, (Bump, Reverse, Turn) - Repeat program. As a challenge, autonomously navigate through an area with obstacles (sensed) This lesson can be integrated with *Mathematics* where concepts such as angles and measurement can be consolidated.
● Design an autonomous program utilizing sensors - Bumper - Gyro	VEX Kits, electronic devices, clawbot, obstacle course Resource: https://www.youtube.com/watch?v=8mQTIr1QrmA	Collaborative Grouping, Teamwork, PBL activity Teams should be able to program the robot to move forward, and when the bumper sensor is touched make a 90 or 180 turn and return to base.

● Program robot to complete 90, 180 and 360 degree turns using sensors - Bumper - Gyro	VEX Curriculum/ STEM Labs	Problem solving, Teamwork, Discussion, Guided Discovery, PBL activity Teams should select leaders who will delegate tasks. Encourage students to implement some type of problem solving strategy when problems in the assembly of the robot occur.
● Problem solve to fix/alleviate any problems with build or coding	VEX Curriculum/ STEM Labs	Teams should be able to discuss, and work collaboratively to solve group issues.

CLASS 4 ● Explain how specific concepts work as it relates to robotics - Mechanical Advantage in a mechanism or system is to achieve desired speed and/or torque.. - Gear ratio - expresses the relationship between a driving gear (gear connected to the input power source, such as a motor), driven gear (the gear connected to the output, such as a wheel or mechanism) in a system.	VEX Curriculum (Mechanisms) handout Resource: https://youtube.com/watch?v=eUeGNe4eDAw https://www.youtube.com/watch?v=odpsm3ybPsA https://www.youtube.com/watch?v=wau5W8Mt7Hw	Collaborative Grouping
● Complete an install of latest version of firmware to the brain of the robot	VEX Software, VEX Curriculum, VEX Guide Resource: https://www.youtube.com/watch?v=4Ti89ErD3VI&t=14s www.vexiq.com/firmware	Collaborative Grouping Before teams work with programming the robot, they will ensure that the latest firmware has been updated to the latest version.
● Demonstrate an understanding of the colour sensor and its functionality	VEX Kits, VEX Curriculum/ STEM Labs, basic robot (Colour sensor, remote control, Cable, Robot brain attached to the robotbase) red and green bristol board Resource: https://www.youtube.com/watch?v=90wyhu-pJ70 https://www.youtube.com/watch?v=VBn7iumcyd4	Collaborative Grouping, PBL activity **Unplugged Activity** - *Red Light, Green Light* - Just as an introductory activity, students can play the game of red light, green light to demonstrate that like the game, the robot sensors understands the difference in colours. Teams will work with the basic build of the robot to observe and understand the functionality of the colour sensor. To test the functionality of the colour sensor,

		teams will attach the sensor to the robot base, and then use red and green pieces of bristol board to demonstrate that the robot will move forward when the green card is shown; and will stay put when the red card is shown.
• Design a basic program using colour sensor	VEX Kits, VEX Curriculum/ STEM Lab, VEX Coloured blocks, Electronic devices	Collaborative Grouping, PBL activity Program the robot to Start, Move Forward (Green), on button Stop (Red)
• Demonstrate an understanding of the distance sensor and its functionality	VEX Kits/ VEX Curriculum/STEM Labs, Robot (distance sensor, robot base, remote control, cable, robot brain) Resource: https://www.youtube.com/watch?v=xRmGcrRxqXQ https://www.youtube.com/watch?v=tU5E2QIsHIo	Collaborative Grouping, Viewing, Problem Solving, PBL activity Teams will build a basic obstacle/wall from the kit blocks, and then attach the distance sensor to the robot base. Drive forward using the remote control towards the obstacle or wall that was built. On sensing the wall, the robot should come to a complete stop.
• Design a basic program using the distance sensor	VEX Robot base, VEX Kit, Electronic Device, Curriculum/STEM Labs	Collaborative Grouping, PBL activity Teams will design a basic program to allow the robot to move autonomously. The program should entail Start, Move Forward

● Build a complex robot adding arm and at least two of the selected sensors - Gyro - Colour - Bumper - Distance	VEX Kits, Electronic devices	Collaborative Grouping, Teamwork, Guided Discovery, Problem Solving, PBL activity In this activity, each group will continue to build a robot adding on the arm. Teams will select which two sensors they will add to their robot.
● Design an obstacle course for the robot build	Paper, pencils	Collaborative Grouping, PBL activity After selecting the two sensors that will be added to the robot, each team will design an obstacle course that will test the functionality of their build.
● Construct an obstacle course for the robot build	Cardboard, bristol board, aluminum foil, tape, string, glue, rulers, sandpaper, PVC piping	Collaborative Grouping, PBL activity This particular activity can be integrated with Visual Arts to ensure its completion. Teams will work together to ensure that the agreed upon design has been constructed and functioning to test the robot build.
● Program robot to carry out a series of functions to complete a specific obstacle course	Completed obstacle course, VEX Curriculum, STEM Lab	Collaborative Grouping, PBL activity

APPENDICES

ABBREVIATIONS USED IN THE DOCUMENT

PBL - Project Based Learning

APPENDIX 1 - Digital Technologies

Key Digital Technologies Terms

Digital Citizen - A member of a worldwide community linked by the internet.

Digital System - A set of elements working together such as hardware and software.

Identity Theft - When a thief steals someone's private information in order to pretend to be that person.

Personal Information - Information that cannot be used to identify you, such as your gender, how many pets you have, etc.

Private Information - Information that can be used to identify you such as your full name, identification number, home address, etc.

Responsibility - An obligation or duty you have to yourself and others

APPENDIX 2 - Computer Language

Key Computer Language Terms

Code - A set of instructions or commands.

Command - A word or code block that tells the computer what to do.

Computer Language - A system of commands (in the form of blocks, words or numbers) that tell a computer how to do things.

Condition - A “true or false?” question that is used to make a decision in a computer program.

Coordinates - The position of an object determined by its x (left to right) and y (top to bottom) value.

Decomposition - The breaking down of a problem into smaller manageable parts.

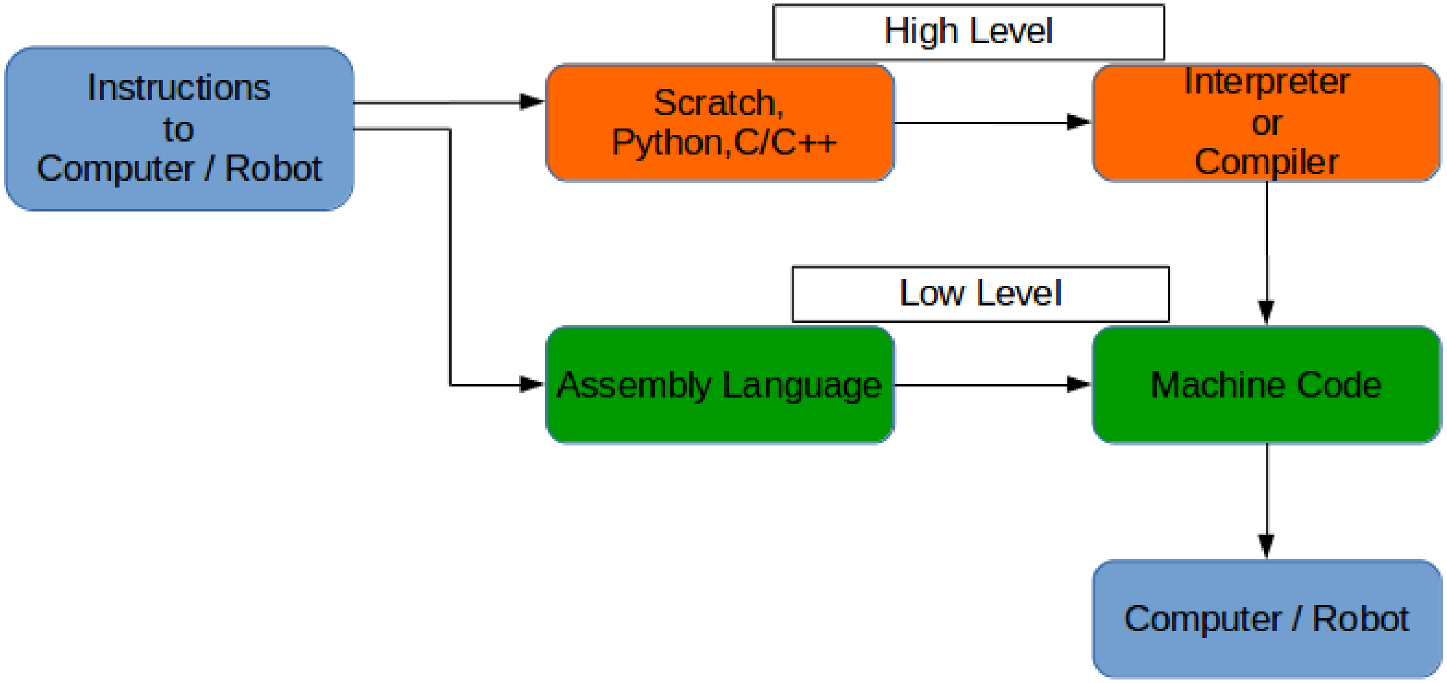
Program - A list of instructions that tell a computer what to do.

Selection - The process of making a decision.

Resources

CS Unplugged - https://classic.csunplugged.org/wp-content/uploads/2015/03/CSUnplugged_OS_2015_v3.1.pdf

Diagrams



APPENDIX 3 - Coding and Programming

Suggested Books

- Sweigart, A. (2016) *Scratch Programming Playground: Learn to Program by Making Cool Games*, William Pollock, No Starch Press Inc.

- Steele, G. (2019) *Coding with Scratch 3.0: Projects Workbook,* Sarah Larter, DK Publishing

- Woodcock, J. (2015) *Coding with Scratch Workbook,* Laura Buller, DK Publishing

- Woodcock, J. & Setford. S. (2016) *Coding in Scratch: Games Workbook,* Sarah Larter, DK Publishing

Exit Ticket

- Computer Programming - https://docs.google.com/presentation/d/1lDZd3ixue2OcDDAcT30mpNaq2CnQfKUUOhN5D5vIpjc/edit?usp=sharing

Key Coding Terms

Animation - Changing pictures quickly to make something appear to move on the screen (for example a sprite dancing).

Backdrop - The picture behind the sprites on the stage.

Block - An instruction in Scratch. Blocks can be joined together.

Bug - A mistake in a program.

Code - A set of instructions or commands.

Command - A word or code block that tells the computer what to do.

Computer Language - A system of commands (in the form of blocks, words or numbers) that tell a computer how to do things.

Condition - A “true or false?” question that is used to make a decision in a computer program.

Coordinates - The position of an object determined by its x (left to right) and y (top to bottom) value.

Costumes - The picture a sprite shows on the stage.

Data - Information - for example, numbers or words.

Decomposition - The breaking down of a problem into smaller manageable parts.

Debug - To remove bugs; to find and fix the errors in a program.

Debugging - The process of identifying and fixing errors in a program when it is not functioning as expected.

Degree - The unit measuring the angle that an object turns.

Duplicate - A simple way to create a copy of a sprite in Scratch.

Event - Something that is triggered on the computer such as a key pressed.

Function - A sequence of code blocks created to do something such as say a greeting in a different language every time the function is run.

If then - A common form of selection in coding where commands are run if something is true.

Input - Data that goes into a program for example from a keyboard.

Iteration - The process of repeating a set of instructions for a specified number of times or until a condition is met.

Library - A collection of sprites, sounds or costumes.

Loop - A sequence of blocks repeated a number of times.

Operator - A block that works something out from data, such as multiplying two numbers together.

Paint Editor - The part of the Scratch interface that is used to draw and edit sprites and backgrounds.

Parallelism - The process of events happening at the same time, either independently or interdependently.

Program - A list of instructions that tell a computer what to do.

Run - To start a program.

Scratch - A visual computer language that uses blocks of code to make a program.

Script - A stack of instructions (blocks) that run in sequential order.

Scripts Area - The area where code blocks are dragged to create programs.

Selection - The process of making a decision.

Sensing group - The set of Scratch blocks that detect when

Sprite - An object (character) that a script can move and change.

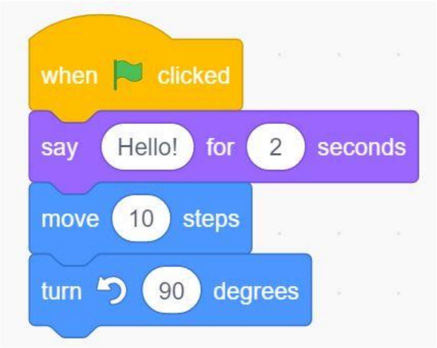
Sprites List - In Scratch 3.0 it is located in the bottom right corner where you select a sprite to add code.

Stage - The area where a Scratch project runs.

Variable - A value or piece of information stored by a computer program.

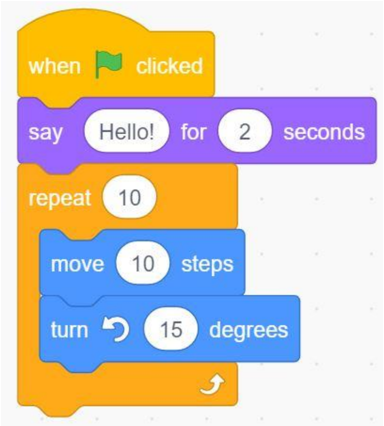
<u>Examples of Code</u>

3.1



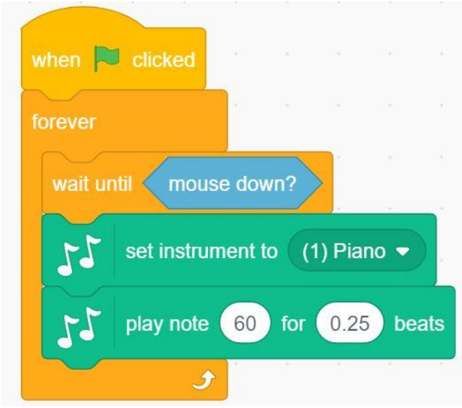
```
when ⚑ clicked
say (Hello!) for (2) seconds
move (10) steps
turn ↺ (90) degrees
```

3.2



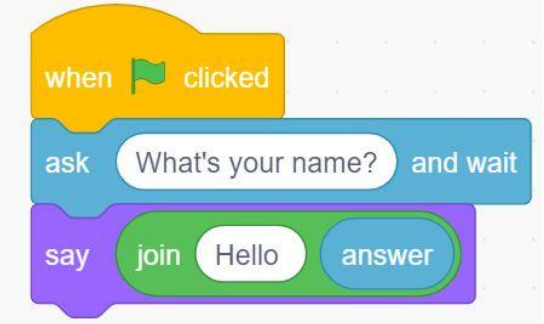
```
when ⚑ clicked
say (Hello!) for (2) seconds
repeat (10)
    move (10) steps
    turn ↺ (15) degrees
```

3.3



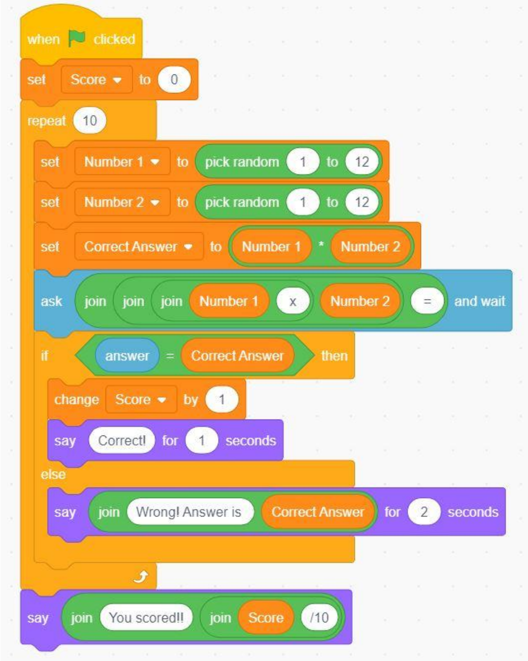
```
when ⚑ clicked
forever
    wait until <mouse down?>
    ♫ set instrument to (1) Piano ▾
    ♫ play note (60) for (0.25) beats
```

3.4



```
when [flag] clicked
ask (What's your name?) and wait
say (join (Hello) (answer))
```

3.5



```
when [flag] clicked
set [Score v] to (0)
repeat (10)
  set [Number 1 v] to (pick random (1) to (12))
  set [Number 2 v] to (pick random (1) to (12))
  set [Correct Answer v] to ((Number 1) * (Number 2))
  ask (join (join (join (Number 1) (x)) (Number 2)) (=)) and wait
  if <(answer) = (Correct Answer)> then
    change [Score v] by (1)
    say (Correct!) for (1) seconds
  else
    say (join (Wrong! Answer is) (Correct Answer)) for (2) seconds
  end
end
say (join (You scored!!) (join (Score) (/10)))
```

APPENDIX 4 - Robotics

Exit Ticket

- Coding & Robotics - https://docs.google.com/presentation/d/1aaey5FEq7RFiP1wu-mQJUunpitUCBgitqTzyAWHa8go/edit?usp=sharing

STEM Terms:

Engineering ***- the application of science and mathematics by which the properties of matter and the sources of energy in nature are made useful to people through products and ideas***

Math ***- the science of numbers, quantities, and shapes and the relations between them***

Innovation (Invention) ***- a new idea, product, process, method, or standard***

Science ***- a system of knowledge covering general truths or the operation of general laws obtained and tested through observation and inquiry***

Technology ***- manner of accomplishing a task using processes, methods, and/or standards***

Robotics ***- the branch of technology that deals with the design, construction, operation, and application of robots***

Key Robotics Terms:

Actuators - ***a mechanical device for moving or controlling something within the environment. This is often based on input from a sensor***

Controller ***- a device or piece of equipment used to operate a machine, vehicle, or system***

Communication Interface - ***a device enabling a robot to communicate with a person or another robot.***

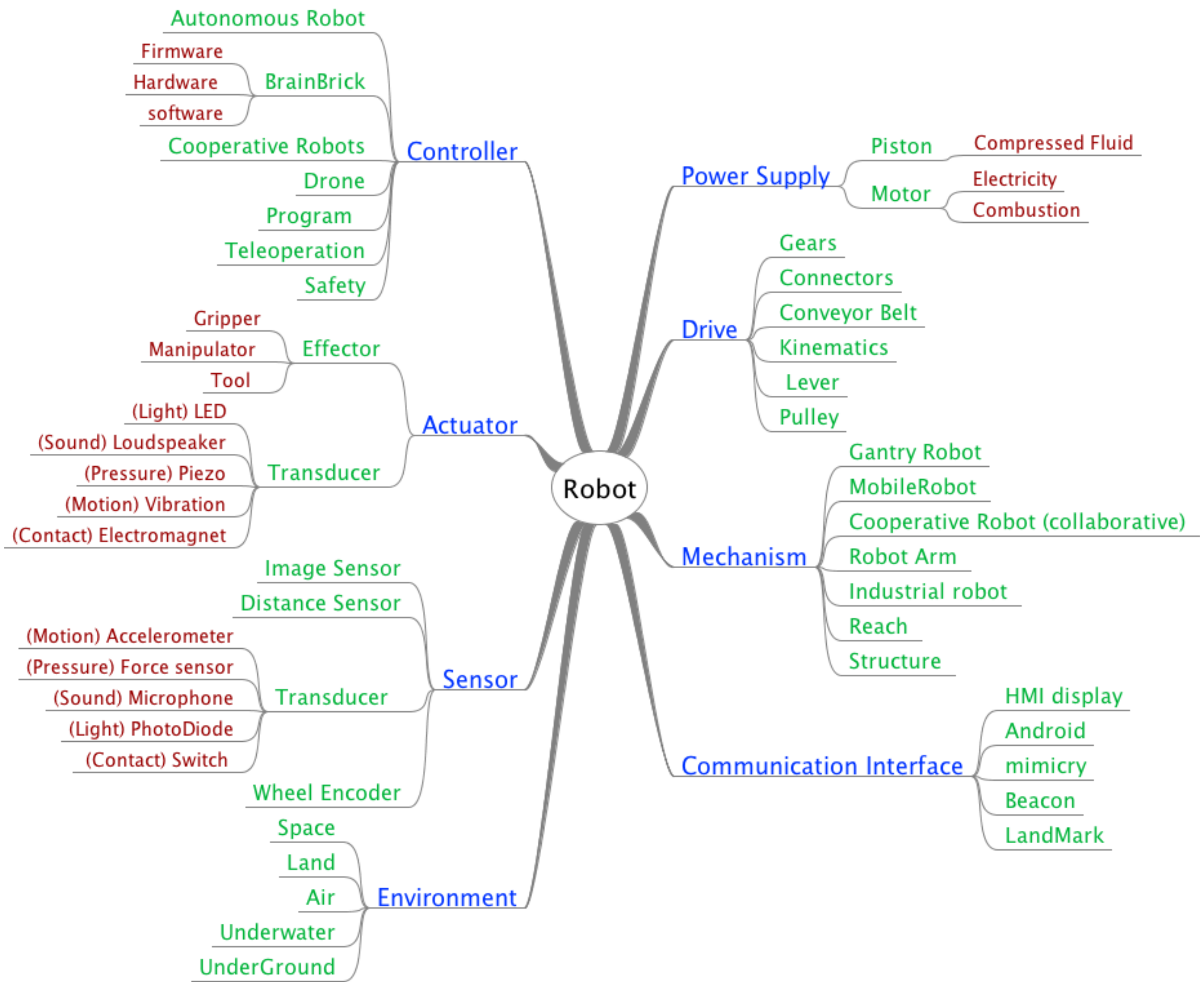
Drive - ***the means for making a machine or machine part move***

Mechanism - ***a mechanical part or group of parts having a particular function***

Power Supply - ***a source or means of supplying energy***

Sensors - ***allow a robot to gather information about its environment. This information can be used to guide the robot's behaviour***

Vocabulary Mind Map



Definitions

Accelerometer - ***a device for measuring acceleration or for detecting and measuring vibrations***

Android - ***a robot with a human appearance***

Autonomous Robot - ***performs behaviors or tasks without regular intervention/instruction from people***

Brain Brick - ***device that determines/controls the autonomous/pre-programmed behavior***

Beacon - ***a strong signal that can be sensed from far away and that is used to guide moving persons/craft***

Connectors - ***mechanical part enabling parts to be joined or fastened together e.g. screw, pin, etc***

Cooperative Robots - ***coordination of systems which consist of multiple robots acting in direct support of persons and/or other robots***

Conveyor Belt - ***a continuous moving band used for moving objects from one place to another***

Distance Sensor - ***a device which detects and conveys information concerning the distance to the nearest object based on the attenuation or reflection of light or sound signals off nearest objects to to those objects***

Drone - ***an unmanned craft (land, air, water, or space) guided by remote control or onboard computers***

Effector - ***a tool that is mounted on a robot, specific to the tasks the robot must perform***

Firmware - ***software programmed into read-only memory, that provides control, monitoring and data manipulation for machines***

Gantry Robot - ***a manipulator mounted onto an overhead system that allows movement across a horizontal plane; also called Cartesian or linear robot***

Gears - ***a part (toothed wheel) that connects the engine of a vehicle or the pedals of a bicycle to the wheels and controls the speed at which the wheels turn***

Gripper - ***an end-effector that provides hand-like grasping capabilities for robot***

Hardware - ***the machines, wiring, and other physical components of an electronic system***

HMI Display - ***provides a control and visualisation interface between a human and a machine***

Image Sensor - ***device that detects and conveys information that constitutes an image by converting variable attenuation of light waves (as they pass through or reflect off objects) into electrical signals***

Industrial Robot - ***an automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes for industrial manufacturing applications***

Kinematics - ***geometry applied to the study of the movement of the mechanisms that form robot structure***

Landmark - ***an object or structure of fixed location that is easy to sense and recognize***

LED Light - ***a device which glows when an electrical signal is applied***

Lever - ***a strong bar that is used, together with a pivot, to lift and move something heavy***

Light Sensor - ***device that converts 'light energy” into an electrical signal***

Manipulator - ***device used to manipulate materials without direct human contact***

Microphone - ***device that converts “sound waves” into an electrical signal***

Mobile Robot - ***robot that moves on land, air, space or water whether Wheeled, Legged, or Propelled***

Motor - ***a rotating machine that transforms electrical energy into mechanical energy***

Piston - ***a disk or short cylinder that slides back and forth inside a larger cylinder; for hydraulics/pneumatics***

Pressure Sensor - ***a device that generates an electrical signal as a function of the pressure imposed***

Program - ***a sequence of coded instructions that can be inserted into/control a robotic mechanism***

Pulley - ***a wheel over which a belt, rope, or chain is pulled to lift or lower a heavy object***

Reach - ***distance from the center of the robot to the fullest extension of the robotic arm (work envelope)***

Robot - ***a machine that can do the work of a person, that works automatically or is controlled by a computer***

Robot Arm - ***mechanical arm with similar movements to a human arm; the arm may be the entire mechanism or may be part of a more complex robot***

Robot Safety ***- implemented by the use of guidelines which ensure that robots are neither deployed improperly nor used carelessly.***

Service Robot ***- performs useful tasks for humans or equipment in non-industrial automation***

Software - ***any set of instructions that directs a computer/robot to perform specific operations***

Speaker ***- a device that converts electrical impulses into sound***

Structure ***- the particles or parts in a substance or body***

Switch ***- a device for making and breaking the connection in an electric circuit (e.g. touch-button)***

TeleOperation - ***electronic remote control of machines***

Transducer ***- device that converts one form of energy to another: typically, variations in a physical quantity, such as pressure or brightness, into an electrical signal, or vice versa.***

Tool - ***a device or implement, especially one held and released, used to carry out a particular function***

Wheel Encoder ***- device for measuring speed at, direction in, and/or distance which a wheel travels***