TSS Primary Computing MTP 2023-2024
Year 3 Block 2 – Secret Codes

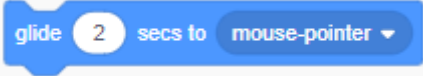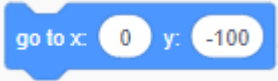| Week	Key Targets and Learning Objectives	Key Activities	Key Vocabulary
1	• Know that ciphers are a way of making sure that information stays secret. • Know how to write and decode messages using very simple code, including converting letters to numbers (1= a 2=b, etc.).	• How long have computers existed? • Look at common items and how long they have been around for. • Discuss when computers were invented and for what purpose. • Introduce the word – cipher and discuss meaning in terms of privacy. Discuss why we might want to keep things private. • Introduce a simple number/letter cipher and decode a simple message. • Practice writing names and composing a simple message in code. • Give children a choice of whether to devise their own code or solve coded activities. • Revisit the main idea and vocab for the lesson.	• code • cipher • secret • private
2	• Follow, understand, edit and correct linear algorithms. • Identify the inputs to an algorithm. • Know how to make a change within a block of code to achieve desired outcomes in programs, such as changing the number of steps a sprite moves. • Predict the outcome of a change to an algorithm that is presented as a sequence of steps. • Outline the benefits of working with others when creating programs.	• Review the ideas of ciphers and codes • Introduce Scratch Scratch - Imagine, Program, Share (mit.edu) • Make sure the children understand that the blocks of code can make the sprite do different things. • Make sure the children understand the following vocabulary: Block Palette, Script Area, Sprites Area, Stage, Sprite • Demonstrate how to put a simple sequence together. • Clarify that the blocks of code are the inputs and that the actions are the outputs. • Change the blocks – what was the input/output? • Allow the children to work in pairs and experiment. • Children to show another pair their ideas. • Discuss blocks of code used and what they have discovered.	• Block Palette • Script Area • Sprites Area • Stage • sprite • input • output • program
3	• Follow, understand, edit and correct linear algorithms. • Identify the inputs to an algorithm. • Know how to make a change within a block of code to achieve desired outcomes in programs, such as changing the number of steps a sprite moves. • Predict the outcome of a change to an algorithm that is presented as a sequence of steps.	• Introduce the Code.org login • Share the slides for lesson 3 (code.org) • Discuss the similarities and differences to what they were doing on Scratch last lesson. • Identify the sprites and programming area. • Allow the children to run through activities to level 9 • Introduce the idea of predicting.	• algorithm • program • sequence • bug • debug • prediction

	• Outline the benefits of working with others when creating programs.	• Review learning and vocabulary	
4	• Follow, understand, edit and correct linear algorithms. • Understand that efficient algorithms are concise. • Know how to make a change within a block of code to achieve desired outcomes in programs, such as changing the number of steps a sprite moves. • Know how to create programs with more than one algorithm running at the same time. • Know how to develop a program that contains more than one object, including a static object. • Know how to develop a program that includes code to reset objects to their original state (initialisation). • Outline the benefits of working with others when creating programs. • Understand that programmers use their mistakes to inform the programs that they create. • Know how to test and debug programs so that it will run and will produce the desired output.	• Review learning from previous week. • Display code that is not concise and discuss how this can be made shorter. • Demonstrate on Scratch how to change the backdrop and add another sprite – just make sure to encourage 1 more sprite and discuss why multiple sprites are not a good idea. • Demonstrate that the sprites have to be programmed separately. • Look at move and glide blocks and discuss the difference. • Look at conversation blocks in the Looks section • Introduce the wait block • Introduce the word initialisation and discuss. • Give children a copy of the conversation code and ask them to choose a backdrop and choose 2 different sprites and change the conversation code in any of the following ways: • alter the spoken script within the 'say' blocks • alter the timings • change the background • change one or both sprites • add a new sprite • extend the conversation. • If time allows share some of the projects • Children can choose to work in pairs or independently.	• Initialisation • sprite • backdrop • conversation
5	• Know that many tasks can be divided into smaller sections to make them easier to follow and to edit. • Know how to develop programs that include code to reset objects to their original state (initialisation). • Know how to create a program with more than one algorithm running at the same time. • Know how to develop programs that contain more than one object, including a static object.	• Review learning from the previous lesson. • Use Lesson 4 from Code.org • Show https://youtu.be/RUZTRNakV9c and discuss the ideas of bugs and debugging in an algorithm. • Get the children to work through Skills levels 2-7 • Review the idea of a prediction before the children work on the levels 8-10 • Children to work independently. • Deal with any difficulties as they occur.	• Bug • debug • code • program • algorithm • step • run

	• Know how to create programs to produce an output from an input device. • Outline the benefits of working with others when creating programs. • Understand that programmers use their mistakes to inform the programs that they create. • Know how to test and debug programs so that they run and produce the desired output.		
6	• Know that many tasks can be divided into smaller sections to make them easier to follow and to edit. • Know how to develop programs that include code to reset objects to their original state (initialisation). • Know how to create a program with more than one algorithm running at the same time. • Know how to develop programs that contain more than one object, including a static object. • Know how to create programs to produce an output from an input device. • Outline the benefits of working with others when creating programs. • Understand that programmers use their mistakes to inform the programs that they create. • Know how to test and debug programs so that they run and produce the desired output.	• Review the learning from last week. • Remind the children how to change a background/backdrop on Scratch Scratch - Imagine, Program, Share (mit.edu) • Explain that the task is to create a game where the ball goes into the net without being saved. • Share this block: `glide 2 secs to mouse-pointer` and discuss what they think the input device is. What is the output from this block? • How many sprites are there? • What does each sprite do? • Children to create, in pairs, a football program to demonstrate the skills learnt. • Children to share and save their work. • Share some of the ideas with the rest of the class. Do any lines of code need to be changed?	• Sprite • code • bug • debug • program • output • input
7	• Follow, understand, edit and correct linear algorithms. • Understand that efficient algorithms are concise. • Know how to make a change within a block of code to achieve desired outcomes in programs, such as changing the number of steps a sprite moves. • Know how to create programs with more than one algorithm running at the same time.	• Review the learning from last week. • Introduce the idea of initialisation. • Share this block with the children: `go to x: 0 y: -100` Demonstrate what this does • Get the children to add this block to their algorithms from last week to return the sprites to a starting position.	• All vocabulary previously covered.

	• Know how to develop a program that contains more than one object, including a static object. • Know how to develop a program that includes code to reset objects to their original state (initialisation). • Outline the benefits of working with others when creating programs. • Understand that programmers use their mistakes to inform the programs that they create. • Know how to test and debug programs so that it will run and will produce the desired output.	• Children to work independently to create a 'game' that has the following criteria: • more than 1 programmed sprite • the initialisation block • an input device (keyboard or mouse) • movement of the sprites • a backdrop/background • Share ideas with the class	