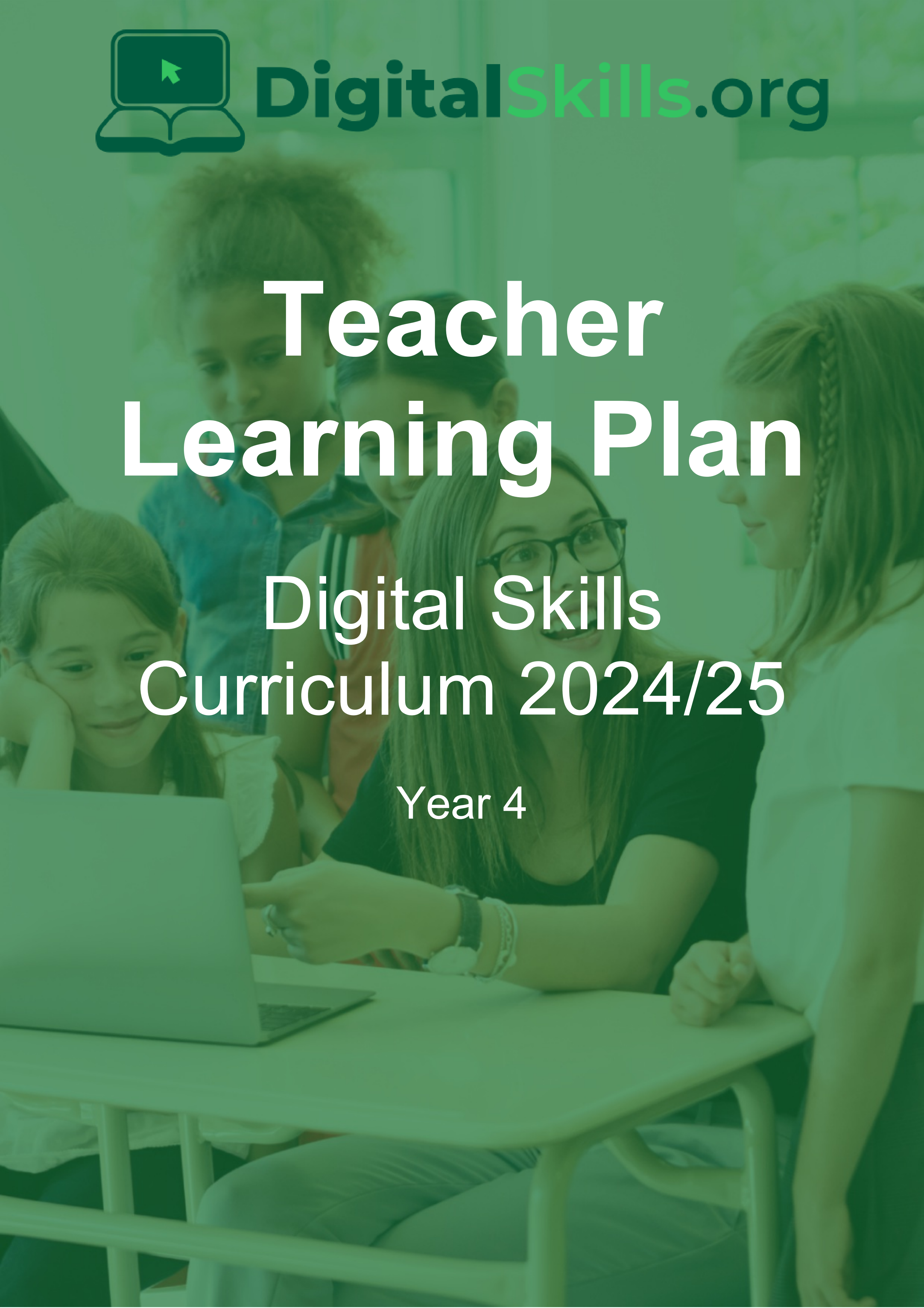


DigitalSkills.org

Teacher Learning Plan

Digital Skills Curriculum 2024/25

Year 4

Table of Contents

How to Use This Learning Plan

This learning plan provides an overview of all the modules available for Year 4, including their units, learning goals, and outcomes. Each module is designed to support both new and experienced teachers with easy-to-follow, step-by-step lessons.

Lesson Types

There are two types of lessons in the Digital Skills Curriculum:

- □ **Teacher-Led Lessons** – The teacher directs and leads students through the lesson, guiding them through the activities and discussions.
- □ **Teacher/Student-Led Lessons** – Teachers can choose to lead the lesson, or students can follow the step-by-step instructions to work through it independently.

Younger students require a fully guided approach, while older students often benefit from working at their own pace with teacher support as needed.

Flexible Curriculum Approach

Teachers have the flexibility to choose the modules that best fit their class needs. While there are enough lessons to cover a full school year, it is not necessary to complete all the modules. This allows teachers to tailor the learning experience to their students while ensuring they meet their educational goals.

Student Access

Students log into the platform to access their lessons. They can follow the step-by-step instructions independently, or teachers can lead the lesson as needed.

Getting Started

1. **Review the Learning Plan:** Each module includes an overview of its goals, learning outcomes, lesson structure, and required resources. Start by familiarising yourself with the curriculum’s scope.
2. **Plan Your Lessons:** Every lesson includes step-by-step guidance, accessible from your teacher dashboard. Adjust the pacing and delivery method based on your students' needs.
3. **Check Required Equipment:** Most lessons only require a laptop, Chromebook, or tablet. Some modules may include additional materials like microbits or LEDs. The required equipment is listed at the start of each module and each individual lesson.
4. **Support Student Learning:** Encourage students to work through the lessons. No prior coding experience is required—teachers can learn alongside their students.
5. **Use Assessments:** Each lesson includes a multiple-choice quiz to help assess student understanding and track progress.
6. **Need Help?:** We're always happy to answer your questions and give advice. You can contact our team at info@digitalskills.org or +44 020 4600 8710.

Module: My First Coding Adventure

This module is designed to introduce students to the world of coding. The first week focuses on the basics of coding, using everyday examples and an instructional game to reinforce the concept. The second lesson introduces Scratch, a coding platform, and guides students through creating their first project. The second week builds on this knowledge, teaching students how to create a dance party animation with a cat and a bat. Teachers should encourage active participation, creativity, and practice throughout The module.

Duration	**Equipment**
2 weeks	Students can use any of these devices: • Chromebook/Laptop/PC • iPad/Tablet
Module Goals	**Module Outcomes**
1. Understand the fundamental concept of coding and its real-world applications. 2. Gain proficiency in using Scratch, a visual programming language, for creating interactive projects. 3. Develop basic coding skills such as creating loops, controlling sprite movements, and changing code numbers. 4. Create a simple animation project using Scratch, demonstrating the ability to add characters and control their movements. 5. Develop an understanding of the importance of precision in coding and the role of practice in mastering coding skills.	1. Understand and explain the concept of coding as giving step-by-step instructions to a computer. 2. Identify and discuss three everyday devices that use coding. 3. Navigate and utilise Scratch to create a basic project, including making a sprite move, adding a loop, changing the numbers, and adding a backdrop. 4. Create a dance party animation with a cat and a bat using Scratch, demonstrating the ability to add characters, control their movements, and add a backdrop. 5. Understand and apply the concept of each sprite having its own code and the importance of practice in mastering coding.

Week 1

Lesson: Introduction to Coding

□ Beginner	□ 20 mins	□ Teacher led

For the 'Introduction to Coding' lesson, begin by explaining coding as giving a computer step-by-step instructions, akin to making a sandwich. Utilise the provided video to further illustrate the concept of coding and its applications. Encourage students to identify three household items that use coding. Reinforce the importance of precise instructions through a game. Conclude by summarizing the key points of the lesson and introducing Scratch, a platform for creating interactive stories, games, and animations, which will be used in the next lesson. Encourage questions and active participation throughout the lesson to foster a conducive learning environment.

Learning Goals	Learning Outcomes
• Understand the basic concept of coding and its role in operating digital devices. • Recognize the presence and application of coding in everyday life. • Develop an understanding of the importance of precise and correct instructions in coding. • Engage in critical thinking by identifying devices in their homes that use coding. • Develop an interest in coding through interactive activities and games. • Prepare for future lessons on coding, specifically using platforms like Scratch.	• Students will be able to define coding and understand its role in everyday life. • Students will be able to identify at least three household items that use coding. • Students will understand the importance of giving precise and correct instructions in coding. • Students will be able to participate in a game that demonstrates the concept of giving instructions to a computer. • Students will be able to discuss and ask questions about the concepts learned in the lesson. • Students will show interest and anticipation for learning more about coding in the next lesson using Scratch.

Lesson: Scratch Tutorial

□ Beginner	□ 40 mins	□ Teacher led	□ Student Quiz

In this Scratch Tutorial lesson, teachers will guide students through the basics of coding using the Scratch platform. Students will learn how to create their own games, animations, and interactive stories. Teachers should create an atmosphere of fun and excitement, sparking curiosity about learning Scratch. The lesson includes watching a short introductory video, navigating the Scratch website, and adding code to make a sprite move, turn, and repeat actions. Students will also learn how to add a backdrop to their projects and experiment with changing code numbers. The lesson concludes with a reflection on what they've learned and what they'd like to learn next.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• Students will understand the basics of coding and its importance in creating games, animations, and interactive stories. • Students will be able to navigate and use the Scratch platform effectively. • Students will learn how to create and control a sprite using various coding blocks in Scratch. • Students will understand how to use loops and control structures to create repeated actions in Scratch. • Students will learn how to add and manipulate backdrops in Scratch to enhance their projects. • Students will develop an attitude of exploration and creativity in coding, leading to the creation of unique and personalized projects.	• Students will be able to navigate the Scratch website and understand its basic functionalities. • Students will be able to create and manipulate a sprite using Scratch's block-based coding system. • Students will be able to implement motion and control blocks to make their sprite move and turn. • Students will be able to use a loop block to make their sprite perform actions repeatedly. • Students will understand how changing the values in their blocks affects the behavior of their sprite. • Students will be able to add and change the backdrop of their Scratch project to enhance the visual appeal.

Week 2

Lesson: A Dancing Cat and Bat

□ Beginner	□ 30 mins	□ Teacher/Student led	□ Student Quiz	□ Student Challenge

This lesson introduces students to the basics of coding using Scratch, a visual programming language. Teachers should familiarize themselves with the Scratch interface and the steps to create a new project. The lesson involves adding sprites, writing simple code to make them move, and controlling the program with start and pause functions. Teachers should guide students through each step, ensuring they understand the purpose of each block of code. Encourage creativity when students choose their sprites and backdrop. Emphasize the concept of each sprite having its own code and the importance of practice in mastering coding.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• Students will understand how to create a new project in Scratch and navigate its interface. • Students will learn how to add sprites from the Scratch library and control their movements using code. • Students will develop the ability to start, pause, and control the execution of their program. • Students will gain knowledge on how to add and manipulate multiple sprites in a single project. • Students will learn how to add and change the backdrop of their Scratch project. • Students will develop a positive attitude towards coding and understand its creative potential.	• By the end of the lesson, students will be able to create a new project in Scratch. • Students will be able to add and control sprites from the Scratch library, as demonstrated by making a cat and bat dance. • Students will be able to write and implement simple code to control the movements of the sprites. • Students will understand and apply the concept of pausing and starting a program. • Students will be able to add a backdrop to their Scratch project, enhancing the visual appeal of their program. • Students will demonstrate their understanding of the lesson by creating their own dance party with different sprites and movements.

Module: Coding and Creative Projects

This module guides students through the basics of coding using Scratch, with a focus on creativity and project-based learning. Each week, students will create a new project, learning key coding concepts and Scratch tools along the way. Teachers should ensure students understand each step before progressing, encourage creativity and experimentation, and provide assistance as needed. The module concludes with a group project, fostering teamwork and idea generation.

Duration	Equipment
6 weeks	Students can use any of these devices: • Chromebook/Laptop/PC • iPad/Tablet
Module Goals	**Module Outcomes**
1. Master the basics of Scratch, including creating new projects, uploading sprites, and using code blocks. 2. Understand and apply the concept of sequencing in coding, demonstrated through the creation of a traffic light sequence. 3. Develop skills in creating interactive games, such as 'Frog Frenzy', and digital art, including character creation and banana stamping. 4. Gain proficiency in advanced Scratch features, including animation of GIFs, creation of virtual pets, and development of an interactive fish tank. 5. Enhance creativity, teamwork, and project management skills through brainstorming and execution of a group Scratch project.	1. Develop a traffic light sequence in Scratch, demonstrating understanding of sequencing and sprite manipulation. 2. Create a simple interactive game in Scratch, incorporating elements of random positioning, scoring, and visual effects. 3. Utilise Scratch tools to create and modify a digital character, demonstrating proficiency in colour manipulation, reshaping, and adding shapes. 4. Construct a banana stamp in Scratch, demonstrating control of the stamp with the mouse, and ability to change its colour, appearance, and clear the canvas. 5. Create a virtual pet in Scratch, demonstrating skills in sprite manipulation, size and colour alteration, and coding for movement and sound. 6. Develop an interactive fish tank in Scratch, demonstrating understanding of mouse-following sprites, clone creation, and randomised clone attributes. 7. Create GIF animations in Scratch, demonstrating proficiency in GIF uploading, animation coding, and exploration of effects and sound. 8. Brainstorm and initiate a simple Scratch project in a group setting, demonstrating creativity, teamwork, and application of Scratch skills learnt.

Week 1

Lesson: Traffic Light Sequence

□ Beginner	□ 40 mins	□ Teacher/Student led	□ Student Quiz	□ Student Challenge

In this lesson, students will learn about sequences in coding by creating a traffic light sequence on Scratch. They will start by creating a new Scratch project and uploading a traffic light sprite. The lesson will then introduce the concept of sequences, using the traffic light as an example. Students will then program the traffic light to display red, green, and orange lights in sequence, each for a specific duration. They will also learn how to make the sequence repeat indefinitely. Finally, they will be challenged to modify their program so that the sequence repeats only a specific number of times.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• Understand and apply the concept of sequencing in computer programming. • Develop skills in using Scratch to create and modify a project. • Learn how to upload and utilize sprites in a Scratch project. • Apply coding skills to create a functioning traffic light sequence. • Understand and apply the concept of loops in coding to make the traffic light sequence repeat. • Develop problem-solving skills by modifying the code to change the sequence repetition.	• Students will be able to create a new Scratch project and upload a specific sprite to it. • Students will understand the concept of a sequence in coding and be able to apply it in a practical context. • Students will be able to program a sprite to change costumes at specific intervals, demonstrating an understanding of the 'wait' command. • Students will be able to make a sequence of commands loop indefinitely, demonstrating an understanding of the 'forever' block. • Students will be able to modify a sequence to repeat a specific number of times, demonstrating an understanding of the 'repeat' block. • Students will be able to share their completed project, demonstrating the ability to follow instructions to submit work.

Week 2

Lesson: Frog Frenzy

□ Beginner	□ 45 mins	□ Teacher/Student led	□ Student Quiz	□ Student Challenge

This lesson, 'Frog Frenzy', is a step-by-step guide to creating a simple game using Scratch. The lesson begins with an introduction to the game and its rules, followed by a series of steps to create the game. These steps include creating a new project, adding a backdrop and a sprite, coding the sprite to disappear and reappear at random positions, creating a score variable, and adding sound and visual effects. The lesson concludes with a wrap-up, encouraging students to experiment and add their own unique touches to the game. Teachers should ensure students understand each step before moving on to the next.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• Students will understand the basic concepts of coding and game design using Scratch. • Students will learn how to create and manipulate sprites and backdrops in Scratch. • Students will develop skills in using code to control the behavior of sprites, such as making them appear and disappear at random positions. • Students will learn how to create and use variables to keep track of game information, such as scores. • Students will learn how to add sound and visual effects to enhance the game experience. • Students will develop an attitude of experimentation and creativity in coding and game design.	• By the end of the lesson, students will be able to create a new project in Scratch, adding specific sprites and backdrops. • Students will be able to write code to make a sprite disappear and reappear at random positions on the screen. • Students will be able to create a score variable and write code to increment the score each time a sprite is clicked. • Students will be able to add sound effects and visual effects to a sprite when it is clicked. • Students will be able to write code to clear all graphic effects from a sprite. • Students will demonstrate their understanding by creating a complete game in Scratch, incorporating all the skills learned in the lesson.

Week 3

Lesson: Create a character

□ Beginner	□ 40 mins	□ Teacher/Student led	□ Student Quiz	□ Student Challenge

In this lesson, students will learn to use the Scratch website to create a digital character. They will use various tools such as the paint bucket, select, reshape, circle, rectangle, and line tools to modify a cat sprite, learning how to change color, delete elements, resize, reshape, and add shapes. After mastering these skills, students will create their own character from scratch. Teachers should familiarize themselves with the Scratch interface and tools in advance, and be prepared to guide students through the process, offering tips and troubleshooting as needed.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• Students will understand how to navigate and use the Scratch website to create digital art. • Students will learn how to use the costumes editor in Scratch to modify existing sprites. • Students will develop skills in using different graphic design tools such as the paint bucket tool, select tool, reshape tool, circle tool, rectangle tool, and line tool. • Students will be able to apply their knowledge of these tools to change the color, shape, and size of different parts of a sprite. • Students will learn how to add new shapes to a sprite, such as circles, squares, and lines. • Students will demonstrate their understanding and skills by creating their own character using the tools and techniques learned in the lesson.	• By the end of the lesson, students will be able to navigate the Scratch website and create a new project. • Students will be able to use the costumes editor in Scratch to modify the appearance of a sprite. • Students will be able to use the paint bucket, select, reshape, circle, rectangle, and line tools in Scratch to modify and create digital art. • Students will be able to change the color of a sprite using the paint bucket tool. • Students will be able to delete, resize, and reshape parts of a sprite using the select and reshape tools. • Students will be able to create a new character sprite from scratch using the learned tools and techniques.

Week 4

Lesson: Banana Stamp

□ Beginner	□ 30 mins	□ Teacher/Student led	□ Student Quiz	□ Student Challenge

In this lesson, students will learn how to create a banana stamp using Scratch. They will learn how to control the stamp with their mouse, change its colour and appearance. The lesson begins with an introduction to the project, followed by steps on how to create a new project on Scratch. Students will then add a banana sprite and the pen extension to their project. They will code the banana to follow their mouse and leave a stamp every time it's clicked. They will also learn how to clear the canvas and change the colour and appearance of the banana stamp. The lesson concludes with a congratulatory message encouraging students to continue exploring with coding.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• Understand how to create and control a sprite in Scratch. • Learn how to use the Pen extension in Scratch to create a stamp effect. • Develop the ability to make a sprite follow the mouse pointer in Scratch. • Learn how to use the stamp command to leave a trail of sprite images. • Understand how to use the 'erase all' command to clear the canvas. • Learn how to change the colour and appearance of the sprite with each click.	• By the end of the lesson, students will be able to create a new project in Scratch, demonstrating their ability to navigate the platform. • Students will be able to add and manipulate sprites in Scratch, specifically the banana sprite, demonstrating their understanding of the sprite library. • Students will be able to use the Pen extension in Scratch to create a stamp effect, demonstrating their ability to add and use extensions. • Students will be able to code the banana sprite to follow the mouse pointer, demonstrating their understanding of event-driven programming in Scratch. • Students will be able to code the banana sprite to leave a stamp, change color, and change appearance when clicked, demonstrating their ability to use conditional statements and effects in Scratch. • Students will be able to code the banana sprite to clear the canvas when the space key is pressed, demonstrating their understanding of keyboard events in Scratch.

Week 5

Lesson: Virtual Pet

□ Beginner	□ 40 mins	□ Teacher/Student led	□ Student Quiz	□ Student Challenge

In this lesson, students will create a virtual pet using Scratch, an online coding platform. They will learn how to add and manipulate sprites and backdrops, use code blocks to change the size and colour of their pet, and make it move and make sounds. The lesson is hands-on and interactive, allowing students to experiment with different code blocks and see the results in real time. Encourage students to be creative and explore different possibilities. At the end of the lesson, students should have a basic understanding of how to use Scratch and be able to create their own simple projects.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• Understand the basics of using Scratch, including creating a new project and adding sprites and backdrops. • Learn how to use the 'change size by' block to alter the size of a sprite, and understand how different keys can trigger different actions. • Develop skills in using the 'change color effect by' block to alter the color of a sprite, and understand how to reset effects when the program starts. • Learn how to use the 'play sound' block to add sounds to a project, and understand how to trigger sounds with user interaction. • Understand how to use the 'turn degrees' and 'move steps' blocks to create movement in a sprite, and understand how to use the 'repeat' block to create repeated actions. • Develop creativity and problem-solving skills by exploring how to add additional features to the virtual pet project.	• Students will be able to create a new project on Scratch and add a specific sprite and backdrop to it. • Students will be able to use the 'change size by' block to increase and decrease the size of their sprite using the up and down arrow keys. • Students will be able to use the 'change [color v] effect by' block to change the color of their sprite when the space key is pressed and reset the color effect when the green flag is clicked. • Students will be able to use the 'play sound', 'turn (10) degrees', and 'move (10) steps' blocks to make their sprite run in a circle and play a sound when clicked. • Students will be able to understand and apply the concept of repeating a set of instructions a specific number of times to achieve a desired outcome (in this case, making a sprite run in a circle).

Week 6

Lesson: Fish Tank Clicker

□ Intermediate	□ 45 mins	□ Teacher/Student led	□ Student Quiz	□ Student Challenge

In this lesson, students will learn to create an interactive fish tank using Scratch. They will be guided to create a new project, add an underwater backdrop, and a fish sprite. Students will then learn to code the fish to follow the mouse pointer, create clones of itself when clicked, and make the clones swim around the screen. They will be taught to use random numbers to vary the size and speed of the fish clones. The lesson aims to teach students the basics of coding, including the use of loops, events, and control blocks, while encouraging creativity and experimentation.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• Students will understand the basics of creating an interactive game using Scratch, including adding sprites and backdrops. • Students will learn how to use loops and event blocks to control the behavior of sprites. • Students will develop the ability to make a sprite follow the mouse pointer on the screen. • Students will gain skills in creating clones of a sprite and controlling their behavior independently. • Students will learn how to use random numbers to vary the size and speed of sprite clones. • Students will develop an understanding of how to experiment with code to create unique and fun interactive experiences.	• By the end of the lesson, students will be able to create an interactive game using Scratch, demonstrated by creating a fish tank game where a fish sprite follows the mouse pointer. • Students will be able to use the 'create clone of' control block in Scratch to create multiple copies of a sprite, demonstrated by creating clones of the fish sprite in the game. • Students will be able to use the 'when this sprite clicked' event block in Scratch, demonstrated by making the fish sprite create a clone of itself when clicked. • Students will be able to use the 'if on edge, bounce' block in Scratch to make sprites bounce back when they hit the edge of the screen, demonstrated by making the fish clones bounce back when they hit the edge of the screen.

Week 7

Lesson: GIFtastic

□ Intermediate	□ 40 mins	□ Teacher/Student led	□ Student Quiz	□ Student Challenge

In this lesson, students will learn how to create GIF animations using Scratch. Teachers should familiarize themselves with the Scratch platform and the process of uploading GIFs and animating them. The lesson begins with an introduction to GIFs and how they work. Students will then create a new project on Scratch and upload a provided Earth GIF. They will learn how to animate the GIF using code. The process is repeated with a Homer Simpson GIF, with an added focus on controlling the speed of the animation. The lesson concludes with an exploration phase where students can experiment with other GIFs, effects, and sounds, and a wrap-up summarizing what they've learned.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• Understand the concept of GIFs and how they function as a series of still images played in quick succession. • Learn how to create a new project on Scratch and upload GIFs into the project as sprites. • Acquire the ability to animate GIFs in Scratch by stepping through the costumes or frames. • Develop the skill to control the speed of animation in Scratch using the 'wait' block. • Explore and experiment with adding effects to the GIFs and incorporating sound or music. • Gain confidence in independently searching for and adding new GIFs to future projects.	• By the end of this lesson, students will be able to understand the concept of GIFs and how they work. • Students will be able to create a new project on Scratch. • Students will be able to upload a GIF into their Scratch project and animate it using coding. • Students will be able to adjust the speed of the GIF's animation by modifying the code. • Students will be able to search for other GIFs on the internet, add them to their project, and animate them. • Students will be able to add sound effects and change the looks of the sprites to create more complex animations.

Week 8

Lesson: Brainstorming Blast

□ Advanced	□ 60 mins	□ Teacher led

In this lesson, 'Brainstorming Blast', students will brainstorm ideas for their own Scratch projects. Start by introducing the lesson and demonstrating a simple Scratch project. Divide students into small groups for brainstorming, reminding them of the importance of teamwork. Set a timer for the brainstorming session and encourage students to keep their ideas simple and achievable. After brainstorming, each group will present their project idea and receive feedback from the class. Students will then create their projects in Scratch, with the teacher providing assistance as needed. Finally, conduct a 'Show and Tell' session where each group presents their project to the class.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• Students will understand the process of brainstorming and how it can be used to generate creative ideas for projects. • Students will develop teamwork skills, including active listening, respectful communication, and collaborative decision-making. • Students will apply their knowledge of Scratch programming to conceptualize and plan a simple project. • Students will gain experience in presenting their ideas to a group and giving and receiving constructive feedback. • Students will develop problem-solving skills as they implement their project ideas in Scratch. • Students will reflect on their learning process, recognizing how their original ideas evolved through implementation and feedback.	• By the end of the lesson, students will be able to effectively brainstorm and develop a simple project idea for Scratch. • Students will be able to work collaboratively in groups, demonstrating effective communication and decision-making skills. • Students will be able to present their project ideas clearly, explaining the concept, sprites, backdrops, and envisioned final product. • Students will be able to provide and receive constructive feedback, and incorporate feedback into their project plans. • Students will be able to create a basic project in Scratch, demonstrating their understanding of the basic Scratch blocks. • Students will be able to present their completed projects, explaining their original idea, implementation process, changes made, and what they learned from the process.

Module: Introduction to Game Design

This module introduces students to game design using MakeCode Arcade. Teachers should familiarise themselves with the platform and guide students through creating various games, including 'Avoid the Enemy', 'Cherry Collection', and 'Beat The Clock'. Encourage creativity and exploration, and ensure students understand key concepts like sprite creation, game loops, and collision detection. The module culminates in a group project where students brainstorm and create their own games.

Duration	**Equipment**
8 weeks	Students can use any of these devices: • Chromebook/Laptop/PC • iPad/Tablet
Module Goals	**Module Outcomes**
1. Master the use of MakeCode Arcade for game design, including sprite creation, control, and effects. 2. Develop skills in creating dynamic gameplay through sprite interaction and game update loops. 3. Understand and implement game elements such as scoring systems, timers, and sound effects. 4. Apply creativity and problem-solving skills in designing and coding unique arcade games. 5. Work collaboratively to brainstorm, receive feedback, and execute a group game project.	1. Utilise MakeCode Arcade to design and control a sprite, and add effects. 2. Create an 'Avoid the Enemy' game using MakeCode Arcade, including sprite interaction and dynamic gameplay. 3. Develop a 'Cherry Collection' game with character movement, random food sprite generation, and sound effects. 4. Design a 'Beat The Clock' game with a timer, scoring system, and sprite movement and interaction. 5. Create a maze game with character control, maze design, camera control, and game-winning conditions.

Week 1

Lesson: Introducing MakeCode Arcade

□ Beginner	□ 60 mins	□ Teacher/Student led	□ Student Quiz	□ Student Challenge

This lesson introduces MakeCode Arcade, a tool for creating arcade games. Teachers should familiarise themselves with the MakeCode Arcade interface and its features, including the code editor, simulator, and toolbox. The lesson guides students through creating a new project, designing a sprite, controlling the sprite's movements, keeping the sprite on screen, and adding effects. Teachers should encourage students to experiment with different effects and explore the potential of MakeCode Arcade.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the basic functionality and usage of MakeCode Arcade. 2. Develop skills in using the MakeCode Arcade code editor. 3. Create and customise a sprite in MakeCode Arcade. 4. Implement controls to move a sprite and keep it within the screen boundaries. 5. Apply and experiment with different effects in MakeCode Arcade.	1. Understand the basic features and functions of MakeCode Arcade. 2. Utilise the MakeCode Arcade code editor effectively. 3. Create and name a new Arcade project on the MakeCode website. 4. Generate and design a sprite character using code. 5. Implement code to control sprite movement and keep it within the screen boundaries. 6. Add and modify effects to the sprite character. 7. Explore further possibilities in game creation using MakeCode Arcade.

Week 2

Lesson: Avoid the Enemy

□ Beginner	□ 60 mins	□ Teacher/Student led	□ Student Quiz	□ Student Challenge

Prepare to guide students through creating a game called 'Avoid the Enemy' using MakeCode Arcade. They'll learn to create and control sprites, and make them interact. The lesson involves creating a new project, designing a player sprite, enabling sprite movement, ensuring the sprite stays on screen, creating enemy sprites, setting enemy positions, and making the enemy chase the player. Celebrate their achievement in creating a dynamic game.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Develop understanding of coding principles through creation of a game. 2. Gain proficiency in creating and controlling sprites in a game environment. 3. Learn to implement game dynamics using update loops. 4. Understand and apply the concept of sprite interaction within a game. 5. Enhance problem-solving skills by debugging and improving game functionality.	1. Create a new Arcade project using the MakeCode Arcade website. 2. Design and control a player sprite in the game. 3. Ensure the player sprite stays within the screen boundaries. 4. Create enemy sprites that appear every 5 seconds. 5. Program enemy sprites to chase the player sprite around the screen.

Week 3

Lesson: Cherry Collection

□ Beginner	□ 60 mins	□ Teacher/Student led	□ Student Quiz	□ Student Challenge

This lesson will guide students through creating a game using MakeCode Arcade. They will learn to create a character sprite, move it around the screen, and collect cherries for points. The lesson includes coding for sprite creation, movement, random placement of cherries, and sound effects. Students will need access to the MakeCode Arcade website. Encourage them to experiment with different sprites and sounds to personalise their game.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Develop skills in using MakeCode Arcade to create a game character and manipulate its movements. 2. Understand and apply the concept of sprites in game development. 3. Implement random generation of game elements on the screen. 4. Apply collision detection to enable interaction between game characters and elements. 5. Incorporate sound and scoring system to enhance game interactivity.	1. Create a new project in MakeCode Arcade. 2. Develop a player sprite and enable its movement using joystick or keyboard arrow keys. 3. Generate food sprites at regular intervals. 4. Position the food sprites randomly on the screen. 5. Implement a function to collect food sprites and increase the score.

Week 4

Lesson: Beat The Clock

□ Intermediate	□ 60 mins	□ Teacher/Student led	□ Student Quiz	□ Student Challenge

This lesson guides students through creating a timed game using MakeCode Arcade. They'll learn to set up game elements, including player and target sprites, and to control sprite movement. They'll also implement a scoring system and a countdown timer. The lesson involves coding, testing, and debugging, encouraging students to experiment with different settings and codes. Ensure students understand the importance of sprite types and how to use the 'randint' function for random positioning.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Develop skills in creating game elements using MakeCode Arcade. 2. Understand how to set up a timer for a game. 3. Learn to create a scoring system in a game. 4. Gain proficiency in using code blocks for sprite interaction. 5. Enhance problem-solving skills by experimenting with different settings and codes.	1. Create game elements using MakeCode Arcade. 2. Set up a timer for the game. 3. Move game elements using joystick or keyboard arrow keys. 4. Implement a scoring system in the game. 5. Use code blocks to detect when two sprites overlap.

Week 5

Lesson: Arcade Build Battles

□ Intermediate	□ 60 mins	□ Teacher led

Prepare to facilitate a series of build battles where students create coding projects within set time limits. Ensure students understand the time constraints and how to share their projects. The battles will vary in length and complexity, from a 15-minute arcade project, to a 5-minute themed project, and finally a 1-minute character design task.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Develop and apply coding skills to create an Arcade project within a specified time limit. 2. Design and create a unique character in Arcade within a one-minute timeframe. 3. Enhance project management skills by adhering to strict time constraints during project development. 4. Improve communication skills by sharing and presenting created projects to peers. 5. Cultivate a competitive spirit and teamwork through participation in build battles.	1. Create an Arcade project within a 15-minute time frame. 2. Share the created project within a 2-minute time frame. 3. Develop an Arcade project with any theme within a 5-minute time frame. 4. Design a character in Arcade within a 1-minute time frame. 5. Share the designed character within a 2-minute time frame.

Week 6

Lesson: Maze Madness

□ Advanced	□ 60 mins	□ Teacher/Student led	□ Student Quiz	□ Student Challenge

This lesson involves creating a maze game using MakeCode Arcade. Students will learn to create a character, control it with buttons, draw a maze, and set a game goal. They will start by creating a new Arcade project, then create a sprite for their character. They will then add code to control their character's movement and draw a maze. They will set the walls of the maze, ensure the camera follows the sprite, add a goal tile, and finally add code to detect if the sprite reaches the goal.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Develop skills in creating and controlling a character sprite in MakeCode Arcade. 2. Understand and apply the concept of drawing and setting up a maze for character navigation. 3. Learn to set up and implement game goals using different tiles. 4. Gain proficiency in using code to detect sprite overlaps and trigger game outcomes. 5. Enhance problem-solving and creative thinking skills through game development.	1. Create a new Arcade project using MakeCode Arcade. 2. Design and implement a sprite character for the game. 3. Enable character control using buttons or keyboard arrow keys. 4. Draw a maze using the tile map editor and set walls to restrict character movement. 5. Implement camera functionality to follow the character throughout the maze. 6. Designate a goal tile within the maze and program game completion upon reaching the goal.

Week 7

Lesson: Fruit Frenzy

□ Advanced	□ 60 mins	□ Teacher/Student led	□ Student Quiz	□ Student Challenge

In this lesson, students will create an interactive game called 'Fruit Collector' using MakeCode Arcade. They will learn about game loops, character controls, collision detection, and scoring. The game involves a character that collects falling fruits while dodging harmful rocks. The lesson includes steps on creating a new project, designing a player character, setting player controls, creating fruit and rock sprites, setting the number of lives, detecting overlap with fruit and rocks, and wrapping up the game. Teachers should ensure students understand the concepts of game loops and collision detection, and are comfortable with using MakeCode Arcade.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Develop skills in using MakeCode Arcade to create a game. 2. Understand and implement game loops and character controls. 3. Learn and apply concepts of collision detection and scoring. 4. Gain knowledge in creating and controlling game elements such as sprites. 5. Enhance problem-solving skills by adding complexity to the game.	1. Create a game character using MakeCode Arcade. 2. Control character movements and position on the screen. 3. Generate and control falling objects (fruits and rocks). 4. Implement collision detection between character and objects. 5. Track and update game score and lives.

Week 8

Lesson: Game Lab

□ Advanced	□ 60 mins	□ Teacher led

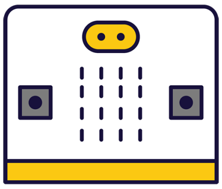
In this lesson, 'Brainstorming Blast', students will brainstorm ideas for their own MakeCode Arcade projects. Start by introducing the lesson and demonstrating a simple MakeCode Arcade project. Divide students into small groups for brainstorming, reminding them of the importance of teamwork. Set a timer for the brainstorming session and encourage students to keep their ideas simple and achievable. After brainstorming, each group will present their project idea and receive feedback from the class. Students will then create their projects in MakeCode Arcade, with the teacher providing assistance as needed. Finally, conduct a 'Show and Tell' session where each group presents their project to the class.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Develop and articulate original ideas for a simple MakeCode Arcade project. 2. Collaborate effectively in small groups to brainstorm and refine project ideas. 3. Present project ideas clearly and constructively, incorporating feedback from peers and teachers. 4. Apply basic MakeCode Arcade blocks to create a simple game or interactive project. 5. Reflect on the process of project creation, identifying learning points and areas for improvement.	1. Brainstorm and develop a simple, achievable idea for a MakeCode Arcade project. 2. Collaborate effectively within a group to discuss and refine project ideas. 3. Present a project idea to the class, explaining the concept, sprites, and tile maps planned for use. 4. Constructively receive and incorporate feedback to improve the project plan. 5. Create a MakeCode Arcade project based on the brainstormed idea, demonstrating basic proficiency in using MakeCode Arcade blocks.

Module: Microbit Basics



Introduce your students to the world of microbits, pocket-sized programmable computers. Guide them through creating a new project, navigating the project editor, and writing their first piece of code. Encourage hands-on learning as they experiment with displaying messages, reacting to button presses, and showing a happy face. In subsequent lessons, they will create emojis, program a dice, measure temperature, learn about compass sensors, create animations, make music, and finally, brainstorm and execute their own microbit project.

Duration	Equipment
7 weeks	Students can use any of these devices: • Chromebook/Laptop/PC Required Equipment: • Microbit
Module Goals	**Module Outcomes**
1. Understand and utilise the basic functions of a microbit, including programming and displaying messages. 2. Create and manipulate simple animations and emojis using the microbit. 3. Program the microbit to perform tasks such as acting as a dice, measuring temperature, and functioning as a compass. 4. Apply coding skills to create interactive narratives and music using the microbit. 5. Develop and execute a unique microbit project, demonstrating creativity, problem-solving, and teamwork.	1. Program a microbit to display messages, react to button presses and show different faces. 2. Develop a microbit program to function as a dice, displaying numbers between 1 and 6. 3. Utilise a microbit to measure temperature and create a temperature-responsive narrative. 4. Calibrate and program a microbit compass sensor to read and display directional data. 5. Create a simple animation on a microbit device using 'show leds' blocks. 6. Program a microbit to play a melody using the 'music.play' function in MakeCode. 7. Brainstorm, develop and present a simple microbit project in a team.

Week 1

Lesson: Meet the Microbit

□ Beginner	□ 60 mins	□ Teacher/Student led	□ Student Quiz	□ Student Challenge

Prepare to introduce students to the world of microbit, a pocket-sized programmable computer. Ensure familiarity with the MakeCode for microbit website, where students will create a new project. Guide them through the project editor, including the microbit simulator, toolbox, and code area. Facilitate the creation of their first code, showing numbers and names, and the use of buttons to display icons. Assist in connecting the microbit to the computer for real-world application. Encourage exploration and experimentation with different blocks and functions.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit

Learning Goals	Learning Outcomes
1. Understand the basic functionality and features of a microbit. 2. Develop skills in creating a new project on the MakeCode for microbit platform. 3. Gain familiarity with the Project Editor interface, including the Microbit Simulator, Toolbox, and Code Area. 4. Acquire the ability to write simple code to display numbers, text, and icons on the microbit. 5. Explore and experiment with different coding blocks to create interactive microbit programs.	1. Identify the key features and functions of a microbit. 2. Create a new project using the MakeCode for microbit website. 3. Understand and navigate the project editor including the microbit simulator, toolbox, and code area. 4. Write and modify code to display numbers and text on the microbit's LED grid. 5. Program the microbit's buttons to display specific messages or icons.

Week 2

Lesson: Emojis

□ Beginner	□ 30 mins	□ Teacher/Student led	□ Student Quiz	□ Student Challenge

Prepare to guide students through creating a new Microbit project on makecode.com. They will learn to program the Microbit to display different faces: asleep, happy, sad, meh, and angry. Students will also design their own face using the 'show leds' block. Ensure they understand how to test their code both in the simulator and on their own Microbit. Familiarise yourself with the accelerometer sensor and its functions.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit

Learning Goals	Learning Outcomes
1. Understand and utilise the Microbit project platform. 2. Programme Microbit to display various emoji faces. 3. Implement button-press functions to change displayed emojis. 4. Use Microbit's accelerometer sensor to trigger emoji changes. 5. Design and implement a custom emoji using the 'show leds' block.	1. Develop a new Microbit project using makecode.com. 2. Program the Microbit to display the 'asleep' face icon. 3. Program the Microbit to display the 'happy' face icon when the A button is pressed. 4. Program the Microbit to display the 'sad' face icon when the B button is pressed. 5. Program the Microbit to display the 'meh' face icon when both A and B buttons are pressed. 6. Test the programmed code in the Microbit simulator. 7. Program the Microbit to display the 'angry' face icon when it is shaken. 8. Design and program a custom face icon to be displayed when the Microbit screen is up. 9. Test the programmed code on a physical Microbit device.

Week 3

Lesson: Microbit Dice

□ Beginner	□ 30 mins	□ Teacher/Student led	□ Student Quiz	□ Student Challenge

Prepare to guide students through creating a new project on the makecode.microbit.org website. They will learn to program the Microbit to detect a shake and display a number between 1 and 6, emulating a dice roll. Ensure they understand how to add and modify code blocks, and how to send their completed code to a Microbit device.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit

Learning Goals	Learning Outcomes
1. Understand and utilise the Microbit project creation process. 2. Programme the Microbit to detect a shake gesture. 3. Display a specific number on the Microbit in response to a shake. 4. Implement a random number generator to simulate a dice roll. 5. Apply the learnt skills to complete a Microbit project and use it as a dice.	1. Develop a new Microbit project using the makecode.microbit.org website. 2. Program the Microbit to detect a shake gesture. 3. Configure the Microbit to display the number 0 upon detecting a shake. 4. Modify the code to display a random number between 1 and 6, simulating a dice roll. 5. Successfully transfer the created code to a physical Microbit device.

Week 4

Lesson: Temperature Tales

□ Beginner	□ 60 mins	□ Teacher/Student led	□ Student Quiz	□ Student Challenge

Prepare for a hands-on lesson involving Microbits and coding. Familiarise yourself with the Microbit coding platform and the process of setting up a new project. Understand how to read temperature from the Microbit's sensor and how to display it. Encourage students to create a narrative that changes with temperature, and guide them in coding this story using 'if' blocks. Finally, facilitate a class presentation where students demonstrate their temperature-responsive stories.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit

Learning Goals	Learning Outcomes
1. Understand and utilise the Microbit coding platform to set up a new project. 2. Develop the ability to read and display temperature data using Microbit's temperature sensor and coding blocks. 3. Create a narrative that varies based on different temperature ranges. 4. Apply logical coding constructs to display different parts of the story depending on the current temperature. 5. Present and explain the temperature-responsive story, demonstrating understanding of how the code manipulates the narrative based on temperature changes.	1. Set up a new project on the Microbit coding platform. 2. Write and test code to read and display temperature from the Microbit's sensor. 3. Create a narrative that changes based on low, medium, and high temperatures. 4. Code the narrative into the Microbit using 'if' blocks to display different parts of the story depending on the temperature. 5. Present the temperature-responsive story, demonstrating how the narrative changes with varying temperatures.

Week 5

Lesson: Compass

□ Intermediate	□ 60 mins	□ Teacher/Student led	□ Student Quiz	□ Student Challenge

In this lesson, students will explore the concept of compass sensors, their functionality, and their application in robotics. They will delve into the workings of a digital compass, focusing on the microbit compass. The lesson will guide them through the calibration process and programming the microbit compass. Finally, they will learn about the various applications of compass sensors in robotics, including navigation and orientation.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit

Learning Goals	Learning Outcomes
1. Develop an understanding of the function and importance of compass sensors in robotics. 2. Comprehend the working mechanism of compass sensors, particularly in relation to the Earth's magnetic field. 3. Gain knowledge about the microbit compass and its features. 4. Learn how to calibrate the microbit compass for accurate readings. 5. Acquire skills to program the microbit compass for specific tasks. 6. Explore various applications of compass sensors in robotic navigation and orientation.	1. Identify the function and importance of compass sensors in robotics. 2. Explain the working mechanism of compass sensors, specifically digital compasses. 3. Describe the microbit compass and its features. 4. Demonstrate the calibration process of the microbit compass. 5. Write code to program the microbit compass for a specific task. 6. Discuss various applications of compass sensors in robotic navigation and orientation.

Week 6

Lesson: Animation

□ Intermediate	□ 60 mins	□ Teacher/Student led	□ Student Quiz	□ Student Challenge

Prepare to guide students through the process of creating an animation using Microbit. Familiarise yourself with creating a new project, adding an event handler, and creating frames for the animation. Ensure you understand the code provided for each step. Be ready to assist students in testing their animations, either on a simulator or a physical Microbit device.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit

Learning Goals	Learning Outcomes
1. Understand and apply the process of creating a new project for animation. 2. Learn to add and utilise an event handler in an animation project. 3. Develop skills to create the first frame of an animation using code. 4. Gain proficiency in creating subsequent frames for a complex animation. 5. Test and evaluate the created animation for functionality and effectiveness.	1. Construct a new project for animation creation. 2. Implement an event handler for specific actions, such as pressing a button. 3. Design the first frame of an animation using 'show leds' block. 4. Develop a second frame for the animation, enhancing the complexity of the pattern. 5. Test the created animation on a simulator or a physical Microbit device.

Week 7

Lesson: Music Maker

□ Advanced	□ 60 mins	□ Teacher/Student led	□ Student Quiz	□ Student Challenge

In this lesson, students will create a new project titled 'Music Maker'. They will learn to add music code using the 'music.play' function, which triggers a melody upon pressing a button. The code can be customised with different melodies and tempos. Finally, students will test their music by pressing the 'A' button on the simulator or on a physical Microbit device.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit

Learning Goals	Learning Outcomes
1. Develop skills to create a new project in a digital environment. 2. Understand and apply the 'music.play' function in coding to create melodies. 3. Modify tempo in music code to alter the speed of the melody. 4. Design a personalised melody using the provided tool. 5. Test and evaluate the functionality of the created music code on a simulator or physical device.	1. Construct a new project with a designated name. 2. Implement the 'music.play' function in the code to generate a melody. 3. Modify the melody and tempo within the 'music.play' function to customise the music. 4. Utilise the 'A' button on the simulator to test the created melody. 5. Download the code onto a physical Microbit device and test the music (optional).

Week 8

Lesson: Microbit Lab

□ Advanced	□ 60 mins	□ Teacher led

Prepare to introduce the concept of Microbit projects, demonstrating a simple LED pattern to inspire creativity. Organise students into small groups for brainstorming, emphasising teamwork and achievable project ideas. Facilitate a feedback session after idea presentations, guiding project simplification if necessary. Assist during project creation, encouraging peer support and discovery sharing. Finally, conduct a 'Show and Tell' session, celebrating student effort and creativity, reinforcing learning objectives and the importance of teamwork.

Students can use any of these devices (and can share if necessary):

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit

Learning Goals	Learning Outcomes
1. Develop creative and achievable project ideas using basic Microbit blocks. 2. Collaborate effectively in small groups to brainstorm, plan and execute a Microbit project. 3. Present project ideas clearly and receive feedback constructively. 4. Apply problem-solving skills to create a Microbit project based on the brainstormed idea. 5. Reflect on the project creation process, discussing changes made, challenges faced, and skills learned.	1. Brainstorm and develop a simple Microbit project idea in a group setting. 2. Present the project idea to the class, explaining the planned LED patterns and inputs. 3. Receive, incorporate, and respond to feedback on the project idea. 4. Create a Microbit project based on the brainstormed idea, using basic Microbit blocks. 5. Present the final Microbit project to the class, explaining the coding process and any changes made during the project creation.