CODING IRELAND

Teacher Learning Plan

Digital Skills Curriculum 2024/25

2nd Year

Table of Contents

How to Use This Learning Plan

This learning plan provides an overview of all the modules available for 2nd Year, 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@codingireland.ie or 01 584 9955.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / 2nd Year / Coding Short Course

Coding Short Course

This course offers a comprehensive introduction to programming, guiding students through fundamental concepts and practical applications. It covers essential coding basics, interactive game development, and web design using Scratch, Arcade, Microbit and other coding platforms. Through hands-on projects, students will build skills in problem-solving and creativity, preparing them for further exploration in computer science.

Duration	Equipment
Classroom hours ~84.66666666666667	Students can use any of these devices: • Chromebook/Laptop/PC • iPad/Tablet Required Equipment: • Microbit • Pen & Paper • Webcam/camera
Learning Goals	**Learning Outcomes**
1. Develop a foundational understanding of coding concepts and their real-world applications. 2. Acquire practical skills in using Scratch and other coding platforms to create interactive projects and games. 3. Explore the principles of web development through HTML, CSS, and JavaScript to build functional webpages. 4. Gain proficiency in programming hardware like Micro:bit for diverse applications such as games and sensors. 5. Foster problem-solving and creative thinking through hands-on coding challenges and project development.	1. Demonstrate an understanding of fundamental coding concepts by explaining the definition and significance of coding in everyday life through practical examples. 2. Create interactive projects using Scratch, including games and animations, by applying skills in sprite manipulation, code blocks, and event handling. 3. Develop functional games and applications with MakeCode Arcade and Microbit, incorporating elements like sprites, variables, and user input controls. 4. Apply HTML, CSS, and JavaScript to build and style basic web pages, integrating interactive features through DOM manipulation and external APIs. 5. Construct programs in Python and JavaScript, utilising variables, loops, and conditional statements to solve specific problems or create functional applications.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Coding Short Course / Module: Coding Basics

Coding Basics

Lesson: Introduction to Coding

□ Beginner	□ 10 mins

Prepare by reviewing the basics of coding, its importance, and how computers work. Facilitate discussions on what coding is and why it matters, using relatable examples like games and apps. Encourage students to identify computers in their homes and practice giving precise instructions through the interactive game provided.

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

- Chromebook/Laptop/PC

- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the fundamental concept of coding as writing instructions for computers. 2. Recognise the importance of coding in everyday technology and problem-solving. 3. Identify the role of computers as tools that execute coded instructions. 4. Explore common household devices that rely on computer instructions. 5. Practise giving precise, logical instructions to achieve desired outcomes.	1. Define coding as the process of writing instructions for computers using a programming language. 2. Explain the importance of coding in creating apps, games, and websites, and its role in problem-solving. 3. Describe computers as electronic devices that process data and follow coded instructions. 4. Identify at least three household items containing computers that can receive instructions. 5. Demonstrate the ability to give precise, ordered instructions through a coding-related game or activity.

Lesson: Scratch Tutorial

□ Beginner	□ 40 mins	□ Student Quiz	□ Student Challenge

This lesson introduces students to Scratch, a coding platform for creating games and animations. Teachers should familiarise themselves with the Scratch website and its functionalities. The lesson guides students through creating a project, removing the default sprite, adding a new sprite, making it move, adjusting values, creating a loop, adding a backdrop, and encourages further exploration. Teachers should be prepared to assist with any technical difficulties and encourage experimentation.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand and navigate the Scratch coding platform. 2. Manipulate sprites by adding, removing, and controlling their movements. 3. Apply basic coding concepts such as loops and event triggers. 4. Modify code blocks to alter sprite behaviour. 5. Explore and experiment with various Scratch functionalities to create unique projects.	1. Identify Scratch as a coding platform for creating games, animations and projects. 2. Navigate and utilise the Scratch website interface. 3. Remove default sprites and add new ones from the sprite library. 4. Implement basic coding blocks to manipulate sprite movement. 5. Modify values within code blocks to alter sprite behaviour. 6. Create a loop within the code to repeat specific actions. 7. Add a backdrop from the library to enhance the visual aspect of the project. 8. Explore and experiment with various code blocks to diversify sprite actions.

Lesson: Paddle Ball Game

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through creating a Paddle Ball Game using Scratch. They'll learn to move sprites, change backdrops, and use sensing blocks. They'll create a new Scratch project, add a paddle and a football sprite, position the ball, make it bounce, control the paddle, make the ball bounce off the paddle, add a backdrop, add a game over line and program the game over. Ensure students understand X and Y coordinates, and how to use the Scratch coding blocks.

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 Scratch to create a simple game. 2. Understand and apply the concept of sprites and backdrops in Scratch. 3. Learn to control sprite movements using mouse input. 4. Implement game logic using conditional statements in Scratch. 5. Understand and apply the concept of X and Y coordinates to position sprites.	1. Manipulate sprites and backdrops in Scratch. 2. Utilise X and Y coordinates to position sprites. 3. Implement code to control sprite movement and interaction. 4. Use sensing blocks to detect sprite collision and mouse position. 5. Create a game over condition using colour detection.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Coding Short Course / Module: Coding Basics

Coding Basics

Lesson: Translate

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

This lesson involves using Scratch to create a language translator. Students will learn how to add the 'Translate' and 'Text to Speech' extensions, create a new project, and add a sprite. They will also learn how to create variables, upload sprites, and write code to translate text into different languages and make the sprite speak the translation. The lesson concludes with the opportunity to add more languages and test the translator.

Learning Goals	Learning Outcomes
1. Understand how to use Scratch to translate text into different languages. 2. Develop skills in creating and managing a new project in Scratch. 3. Learn to use the Translate and Text to Speech extensions in Scratch. 4. Gain experience in creating and using variables in Scratch. 5. Apply knowledge to add more languages to the translation project.	1. Utilise Scratch to create a new project and add specific extensions. 2. Implement the Translate and Text to Speech extensions in a Scratch project. 3. Create and manipulate variables within Scratch to store language and translation data. 4. Use Scratch to translate text into different languages and vocalise the translation. 5. Add and code multiple language options to a Scratch project.

Lesson: Shark Swim

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

In this lesson, students will create a game using Scratch, where a diver navigates a course without touching the edges or encountering a shark. They will learn how to set up a new Scratch project, create a backdrop, add and position sprites, and write code to control sprite movements. They will also learn how to animate sprites using costumes, detect collisions, and create a simple game loop. The lesson concludes with students testing their game and reflecting on their learning.

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 managing a Scratch project. 2. Understand and apply the concept of sprite control and animation using costumes. 3. Gain proficiency in using coding blocks for game mechanics such as collision detection and game loop creation. 4. Learn to use the mouse pointer for sprite movement and control. 5. Develop an understanding of game development concepts and apply them in a practical project.	1. Develop a game using Scratch, incorporating elements such as sprites, backdrops, and costumes. 2. Control sprite movements using mouse pointer and code blocks. 3. Implement collision detection between sprites and specific colours. 4. Utilise costumes to create animation effects within the game. 5. Create a simple game loop, demonstrating understanding of game development basics.

Lesson: Autonomous Car

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through the process of understanding how autonomous cars work. Facilitate the creation of a Scratch project where students will program their own autonomous car, incorporating elements such as car sprites, speed variables, and sensor-driven navigation. Encourage students to experiment with different track designs and speeds, fostering a deeper understanding of autonomous vehicle technology.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the concept and workings of an autonomous car. 2. Develop skills in creating a new Scratch project and manipulating sprites. 3. Learn to use variables and conditional statements in Scratch to control sprite movements. 4. Apply knowledge of sensors in programming an autonomous car to navigate a track. 5. Enhance problem-solving skills by implementing speed control and reverse functions in the autonomous car project.	1. Understand the functioning of an autonomous car and its use of sensors for navigation. 2. Create a new Scratch project and manipulate sprites and backdrops. 3. Program the car to move and navigate using colour detection and conditional statements. 4. Control the speed of the car using variables and keyboard inputs. 5. Implement a reverse function to correct the car's course when it deviates from the track.

Lesson: Pattern Creator

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through an engaging exploration of pattern creation using Scratch. Familiarise yourself with the Scratch interface and pen tool, as well as the process of creating a new project and adding sprites. Be ready to explain the use of variables, loops, and how to manipulate pen colour and size. Encourage students to experiment with different degrees and pen sizes to create unique patterns. Wrap up by reinforcing the importance of practice and creativity in mastering coding.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Master the use of Scratch for pattern creation. 2. Understand and apply the use of variables in creating complex shapes and patterns. 3. Manipulate the pen tool to draw and create unique patterns. 4. Experiment with different degrees and pen sizes to alter pattern outcomes. 5. Apply creativity in coding to produce vibrant and unique patterns.	1. Code a Scratch project to create basic patterns using the pen tool. 2. Implement the use of variables to manipulate pattern creation. 3. Adjust pen colour and size to enhance pattern design. 4. Utilise loops and conditional statements to control pattern formation. 5. Experiment with different variable values to create unique patterns.

Lesson: Attack of the Dots

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare for an interactive lesson where students will create a game using Scratch. They will learn to control a coloured disc, clone attacking dots, and detect the colour of the dots. Ensure students understand how to remix a starter project, make the disc spin, clone the ball, prevent the ball from appearing too close to the disc, make the ball move, detect the colour of the ball, create purple and orange balls, and change the code for the purple and orange balls. Wrap up by congratulating students on their newly acquired skills.

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 Scratch to create an interactive game. 2. Understand how to control a coloured disc using keyboard inputs. 3. Learn to clone game elements and set their behaviour. 4. Master the technique of colour detection for game mechanics. 5. Apply problem-solving skills to prevent game elements from spawning too close to the player.	1. Master the use of Scratch to create an interactive game. 2. Control a coloured disc using keyboard inputs. 3. Clone and manipulate game elements, such as coloured dots, using Scratch code. 4. Implement colour detection to trigger game events. 5. Modify and customise game elements to enhance gameplay.

Lesson: Rocket Lander

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through creating a rocket landing game using Scratch. The lesson involves programming gravity, controlling rocket movement, creating animations for rocket thrust and explosion, and adding a fuel limit for an extra challenge. Ensure students understand the concept of variables and conditions in coding. Encourage creativity and problem-solving as they experiment with their game.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the concept of vertical rocket landing and its challenges. 2. Develop a game using Scratch, simulating a rocket landing scenario. 3. Implement gravity and movement controls in the game using code blocks. 4. Create and use costumes to animate rocket thrust and explosion. 5. Introduce and manage a fuel limit for added complexity in the game.	1. Understand and explain the functionality of the Space X Falcon 9 rocket. 2. Create a basic game in Scratch, including setting up a starter project. 3. Program gravity and booster functions for a rocket sprite in Scratch. 4. Design and implement visual effects such as rocket thrust and explosion in Scratch. 5. Implement controls for rocket movement and landing, including fuel limits and landing conditions.

Lesson: Scratch Platformer

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

In this lesson, students will create a platformer game using Scratch. They will design characters, create platforms, and write code to control character movements. The lesson includes creating a new Scratch project, designing sprites, resizing characters, creating variables, applying gravity, enabling character movement and jumping, adding a trailing effect, adding more costumes to the ground sprite, detecting screen edges, receiving messages, and wrapping up. The lesson is hands-on and encourages creativity and problem-solving.

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 manipulating sprites in Scratch. 2. Understand and apply the concept of variables in game development. 3. Implement control mechanisms for character movement, including gravity and jumping. 4. Utilise broadcasting messages to manage game states and transitions. 5. Enhance game aesthetics through effects like character trailing.	1. Design and create sprites for a platformer game in Scratch. 2. Implement movement controls for a character sprite, including left, right, and jump actions. 3. Apply gravity effect to character sprite using Scratch coding blocks. 4. Create and utilise variables to control game mechanics such as speed, jump height, and gravity. 5. Develop multiple game levels by creating different platform configurations.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Coding Short Course / Module: Coding Basics

Coding Basics

Lesson: First Arcade Project

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

This lesson guides students through creating their first arcade project using MakeCode Arcade. They will learn about the code editor, how to create a new project, add a sprite, choose a sprite from the gallery, move the sprite, draw a tile map, draw walls, make the camera follow the sprite, add projectiles, set their direction and speed, detect overlap, lose a life, and finally, send the code to a handheld device. The lesson is hands-on and interactive, allowing students to learn by doing.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand and utilise MakeCode Arcade for creating games. 2. Manipulate the Code Editor to build and modify game elements. 3. Create and customise sprites for use in a game. 4. Develop a tile map and implement walls for game navigation. 5. Implement game mechanics such as projectiles, sprite movement, and life count.	1. Understand the functions and features of MakeCode Arcade. 2. Use the MakeCode Arcade code editor to create a new project and add a sprite. 3. Manipulate the sprite's movements using the direction buttons in the simulator. 4. Create and edit a tile map, including drawing walls and setting the camera to follow the sprite. 5. Design and implement projectiles, including setting their direction and speed, and programming responses to overlaps with the player's sprite.

Lesson: Monkey Mayhem

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through creating a game using MakeCode Arcade. They will learn to control a character, generate objects at random positions, and collect them for points. They will also add a countdown timer to make the game more challenging. Ensure students understand the concepts of sprites, coordinates, and coding effects. Encourage creativity and problem-solving as they modify the game or create a new one.

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 player sprite in MakeCode Arcade. 2. Understand how to generate food sprites at random positions on the game screen. 3. Learn to implement a scoring system based on sprite interaction. 4. Gain knowledge on adding sound effects to enhance game experience. 5. Master the use of a countdown timer to increase game difficulty.	1. Create and control a player sprite in MakeCode Arcade. 2. Generate food sprites at random positions on the game screen. 3. Collect food sprites for points and implement a scoring system. 4. Add sound effects to enhance game play experience. 5. Implement a countdown timer to increase game challenge.

Lesson: Space Shooter

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through creating a space-themed game using MakeCode Arcade. They will design a spaceship sprite, control its movements, set the number of lives, create and program asteroids, fire rockets, destroy asteroids, and lose lives when hit by an asteroid. Ensure students understand the importance of correct code placement and sprite selection. Encourage them to test their game frequently to ensure it functions as expected.

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 MakeCode Arcade for game creation. 2. Gain proficiency in creating and controlling game sprites. 3. Learn to implement game mechanics such as scoring and lives. 4. Understand how to detect and respond to sprite interactions. 5. Apply coding skills to create a complete Space Shooter game.	1. Design and create a spaceship sprite in MakeCode Arcade. 2. Control the spaceship sprite using arrow keys and prevent it from going off the screen. 3. Set the number of lives for the spaceship. 4. Create and program asteroids to fly in from the right side of the screen. 5. Fire rockets from the spaceship when the A button is pressed.

Lesson: Platform Place

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through creating their first platform game using MakeCode Arcade. The lesson involves understanding the basics of platform games, creating a new project, designing a sprite, programming sprite movements, adding gravity, drawing a map with different elements, programming a jump function, testing the game, and adjusting the game's mechanics. Ensure students understand the code snippets and their purpose in the game's functionality. Encourage creativity in sprite and map design.

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 concept and mechanics of platform games. 2. Create and design a sprite character in a game environment. 3. Implement movement controls for the sprite character. 4. Apply the concept of gravity in a game setting. 5. Design and create a game map with different elements such as ground, danger and goal tiles.	1. Understand the concept of platform games and their mechanics. 2. Create a new project on arcade.makecode.com and design a sprite character. 3. Implement sprite movement controls using code. 4. Apply the concept of gravity to a sprite in a platform game. 5. Design a game map with ground, danger, and goal tiles. 6. Program a sprite to jump and move through the map. 7. Implement game mechanics such as danger tiles and a goal tile.

Lesson: Arcade Build Battles

□ Intermediate	□ 60 mins

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.

Lesson: Galaxy Ghosts

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through creating a space-themed game using MakeCode Arcade. They will learn to create and control a player sprite, generate enemy sprites, and program interactions between them. The lesson includes creating a new project, coding the player and enemy sprites, setting their positions and movements, and programming the game's scoring system and health bar. Ensure students understand each step and encourage them to experiment with their games.

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 space-themed game. 2. Learn to create and control player and enemy sprites, and program interactions between them. 3. Understand how to implement a scoring system and a health bar in the game. 4. Gain knowledge on how to increase game difficulty by increasing enemy speed over time. 5. Develop problem-solving skills by modifying and improving the game.	1. Create and control a player sprite in MakeCode Arcade. 2. Generate enemy sprites and program interactions between them. 3. Implement a scoring system for each enemy sprite hit. 4. Use a health bar to track and display player's health status. 5. Program game over conditions based on player's health status.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Coding Short Course / Module: Coding Basics

Coding Basics

Lesson: Exploring Microbits

□ Beginner	□ 60 mins	□ Student Quiz

Prepare to introduce students to the world of microbits, a pocket-sized programmable computer. The lesson will involve creating a new project on the MakeCode for microbit website, familiarising with the project editor, and writing code to display numbers, names, and icons. Students will also learn to delete code, connect their microbits to their computers, and program their microbits to play music. The lesson concludes with an exploration phase where students can experiment with different blocks from the toolbox.

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. Create a new project using the MakeCode for microbit website. 3. Use the Project Editor to write and simulate code. 4. Program the microbit to display numbers and text on its LED grid. 5. Program the microbit to respond to button presses with specific actions.	1. Identify the functions and capabilities of a microbit. 2. Create a new project on the MakeCode for microbit website. 3. Understand the layout and functions of the Project Editor. 4. Write and execute code to display numbers and names on the microbit. 5. Program the microbit to respond to button presses with specific displays. 6. Connect and download code to an actual microbit device. 7. Compose and program a melody to play on the microbit. 8. Explore and experiment with different coding blocks and functions.

Lesson: Microbit Step Counter

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through creating a Microbit step counter. They'll start a new project on makecode.microbit.org, create and set up a 'steps' variable, and use the accelerometer to detect steps. They'll write code to display the step count and send it to their Microbit. After connecting a power source, they'll secure the Microbit to their person and start walking. They'll adjust the code to count every step and resend the updated code to their Microbit. Caution them to be careful while walking with the Microbit.

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 a basic understanding of Microbit programming and project creation. 2. Learn to create and set up variables in Microbit. 3. Understand the use of accelerometer sensor in Microbit for step detection. 4. Gain skills to display data on Microbit using LEDs. 5. Learn to modify and resend code to Microbit for improved functionality.	1. Develop a new Microbit project using the makecode.microbit.org website. 2. Create and set up a 'steps' variable to record the number of steps taken. 3. Utilise the accelerometer sensor in Microbits to detect and record steps. 4. Display the recorded number of steps on the Microbit using its LEDs. 5. Modify the code to accurately count every step taken, and resend the updated code to the Microbit.

Lesson: Reaction Timer

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students in creating a 'Reaction Timer' project using Micro:bit. They'll start by setting up a new project, then create a welcome message and a countdown. Next, they'll add a random delay to make the game unpredictable. They'll create variables to store time stamps, and finally, record the player's reaction time. Familiarise yourself with the code snippets provided.

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 in creating and managing a new project on the Micro:bit platform. 2. Acquire knowledge on how to create and display messages using code. 3. Understand and apply the concept of countdowns and delays in programming. 4. Learn to create and utilise variables for storing time stamps. 5. Gain proficiency in recording and displaying user interactions in real-time.	1. Develop a new project using the Micro:bit website. 2. Construct a welcome message to display upon powering on the Microbit. 3. Create a countdown sequence with visual cues using code. 4. Implement a random delay function in the game for unpredictability. 5. Create and utilise variables to store time stamps. 6. Record and display player reaction time upon button press.

Lesson: Microbit Fruit and Veg Piano

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to demonstrate the conductivity of the human body and various fruits and vegetables using a Microbit. Gather a Microbit, 4 crocodile clips, and 4 pieces of fruit or vegetables. Familiarise yourself with the Microbit programming interface and the specific code for programming Pins 0, 1, and 2. Ensure you understand how to connect the crocodile clips and test the circuits. Be ready to guide students in connecting the fruit and vegetables to create a musical instrument.

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 concept of electrical conductivity using the human body and various fruits and vegetables. 2. Identify and utilise the components of a Microbit, including its pins and GND. 3. Create and modify a Microbit project using the makecode.microbit.org platform. 4. Program Microbit pins to play different musical notes and display different icons. 5. Test and troubleshoot a simple electrical circuit using a Microbit, crocodile clips, and conductive materials.	1. Identify and gather necessary materials for creating an electrical circuit with a Microbit and fruit or vegetables. 2. Create a new project on the makecode.microbit.org website. 3. Program Pins 0, 1, and 2 on the Microbit to play different notes and display different icons when pressed. 4. Connect crocodile clips to Pins 0, 1, 2 and GND on the Microbit and test the circuit. 5. Attach fruit or vegetables to the crocodile clips and demonstrate the ability to play different notes by touching and releasing each piece.

Lesson: Designing a Microbits Weather Station

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare for this lesson by familiarising yourself with the MakeCode for Microbit platform and the coding language used. Understand the purpose of variables and how they can be initialised and manipulated. Be prepared to guide students through the process of creating a new project, configuring buttons and sensors, creating a 'forever' loop, and testing their program. Encourage reflection on the learning process and potential applications of the skills learned.

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 how to create a new project on MakeCode for Microbit. 2. Learn to declare and initialise variables in a Microbit project. 3. Gain skills in configuring Microbit buttons and sound detection. 4. Develop the ability to create a 'forever' loop and display sensor data. 5. Reflect on the learning process and consider potential applications of Microbit sensors.	1. Develop a new project using MakeCode for Microbit. 2. Declare and initialise variables 'mode' and 'reading' for sensor data display and storage. 3. Configure Button A to set 'mode' to 1 when pressed. 4. Configure Button B to set 'mode' to 2 when pressed. 5. Configure Buttons A and B together to set 'mode' to 3 when pressed simultaneously. 6. Configure the Microbit to switch to mode 4 when a loud sound is detected. 7. Create a 'forever' loop to check the value of 'mode' and display the relevant sensor data. 8. Test the program using a physical Microbit or the simulator on the MakeCode website. 9. Reflect on the learning process, understanding how the different sensors on the Microbit work and potential other projects.

Lesson: Microbit Compass and Thermometer

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students in creating a Microbit project that utilises the compass and temperature sensor. They will learn to create and set variables, program buttons, and use 'if then else' blocks. The lesson involves coding the Microbit to display cardinal directions based on its orientation and temperature readings. Students will also test their code using a simulator before sending it to their Microbit. Ensure familiarity with the makecode.com platform and basic coding concepts.

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 compass and temperature sensor features of the Microbit. 2. Develop proficiency in creating and setting variables in a Microbit project. 3. Apply conditional logic to program Microbit buttons for specific functions. 4. Test and debug code using the simulator before transferring to the Microbit. 5. Interpret and display data from the Microbit's sensors in a user-friendly format.	1. Develop a new Microbit project using makecode.com. 2. Create and set a 'direction' variable to store compass readings. 3. Program the A button to display compass direction (N, S, E, W) based on 'direction' variable. 4. Program the B button to display the current temperature reading. 5. Test and debug the code using the simulator and then deploy it to the Microbit.

Lesson: Microbit Pet

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

In this lesson, students will transform their Microbits into interactive pets. They will use emoji icons and sounds to make the Microbits seem lifelike, programming them to respond to different actions such as shaking, touching, and flipping. Students will create functions for different states of the pet, like happy, sad, hungry, bored, and asleep. They will also learn to use the Microbit's sensors to detect these actions. The lesson involves coding in the Microbit's online editor, testing the code in a simulator, and finally downloading it onto their Microbits.

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 in creating and using functions in Microbit programming. 2. Understand and apply the use of different sensors on the Microbit device. 3. Gain knowledge in programming interactive responses using sound and visual cues. 4. Learn to use random time intervals in programming for unpredictable outcomes. 5. Enhance problem-solving skills by debugging and testing code in a simulator and on a physical device.	1. Program Microbit to display different emoji icons and sounds to simulate pet behaviours. 2. Create a new Microbit project using the provided website. 3. Develop functions such as 'happy', 'feedme', and 'play' to control pet behaviours. 4. Implement gesture controls to interact with the Microbit pet, such as shaking, flipping, and touching the logo. 5. Test the programmed Microbit pet in the simulator and download the code onto a physical Microbit.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Coding Short Course / Module: Intermediate Coding

Intermediate Coding

Lesson: Bat Battle

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

This lesson guides students through creating a game using MakeCode Arcade. They will learn to create and control a player sprite, generate enemy sprites, and program interactions between them. The lesson includes coding for scoring points and ending the game. Teachers should ensure students understand each step before moving on, and encourage experimentation with the code to add new features to the 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 an interactive game. 2. Understand how to create, control, and position player and enemy sprites. 3. Learn to program game interactions such as shooting projectiles and detecting overlaps. 4. Gain knowledge on how to keep score and end the game in MakeCode Arcade. 5. Enhance problem-solving and debugging skills by experimenting with the code and adding new features.	1. Create and control a player sprite in MakeCode Arcade. 2. Generate enemy sprites at random positions. 3. Program interactions between player and enemy sprites. 4. Implement a scoring system for hitting targets. 5. End the game when an enemy sprite hits the player sprite.

Lesson: Monster Battle Arena

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through creating a 'Monster Battle Arena' game using MakeCode Arcade. They will learn to create player-controlled and AI-controlled sprites, implement combat mechanics, health systems, and AI behaviours. Students will also learn to create a new project, design sprites, make the monster move, implement a health system, create a combat system, and determine the winner. Encourage creativity and experimentation with the game's features.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Develop a player-controlled sprite and an AI-controlled monster in a game using MakeCode Arcade. 2. Create and manage a new project in MakeCode Arcade. 3. Implement a health system for player and monster sprites. 4. Develop a combat system where player and monster sprites can inflict damage on each other. 5. Implement a win/lose condition based on the health of player and monster sprites.	1. Create and control a player sprite using MakeCode Arcade, ensuring it moves smoothly within the screen boundaries. 2. Program an AI-controlled monster sprite with randomized movement, simulating intelligent behavior. 3. Implement a health system that tracks and displays the health values of both the player and the monster during the game. 4. Develop a combat system that reduces player health upon collision with the monster and allows the player to shoot projectiles at the monster. 5. Determine the game's winner by programming conditions that end the game when either the player's or monster's health reaches zero.

Lesson: Donut Rush

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

In this lesson, students will create an interactive game called 'Donut Rush' using MakeCode Arcade. They will learn to write code for creating game sprites, handling events like sprite overlaps, and controlling game logic. The lesson involves setting up the game, creating a new project, and defining variables to track the game's state. Students will also learn to create a function, set up the level, create the donuts, and start the game. They will add code to detect when the player sprite overlaps with a donut sprite and to check if the player has collected the target number of donuts. The lesson concludes with a wrap-up and play session.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Develop an understanding of game creation using MakeCode Arcade. 2. Learn to create and manage variables in a gaming context. 3. Understand the concept and application of functions in game development. 4. Gain skills in handling events such as sprite overlaps and controlling game logic. 5. Apply knowledge to create an interactive game with multiple levels and scoring system.	1. Create a new project in MakeCode Arcade. 2. Set up the game by creating a splash screen, setting up variables, and creating a player sprite. 3. Create a function called 'startLevel' to organise the game's code. 4. Set up the level by adding code to the 'startLevel' function, including setting the background colour, displaying a level message, setting the target number of donuts to collect, and starting a countdown. 5. Create multiple donuts using a loop and place them randomly on the screen. 6. Start the game by calling the 'startLevel' function. 7. Collect donuts by detecting when the player sprite overlaps with a donut sprite, increasing the score, destroying the donut sprite, and playing a smile effect. 8. Complete the level by checking if the player has collected the target number of donuts, increasing the level, playing a 'jump up' sound, and starting a new level. 9. Wrap up the game and play it, aiming to collect as many donuts as possible within the time limit.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Coding Short Course / Module: Intermediate Coding

Intermediate Coding

Lesson: The World of the Internet

□ Beginner	□ 30 mins	□ Student Quiz	□ Student Challenge

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 the Internet, its history, and its significance in the modern world. • Comprehend how the Internet works, including the process of sending and receiving information. • Explore the various uses of the Internet, including messaging, information sharing, and accessing websites. • Learn about the structure and purpose of URLs and how they are used to access websites. • Recognize the importance of responsible and respectful behavior when using the Internet. • Appreciate the role of the Internet in connecting people globally and facilitating information exchange.	• By the end of this lesson, students will be able to define what the Internet is and explain its importance. • Students will be able to describe how the Internet works, including how information is sent and received. • Students will be able to identify different ways to use the Internet, such as sending messages and sharing information. • Students will be able to explain how messages are sent over the Internet, using the concept of 'data packets'. • Students will be able to identify different platforms for sharing information on the Internet, including websites, blogs, and social media. • Students will be able to understand and explain what a URL is and how it is used to access websites.

Lesson: How Does the Internet Work?

□ Beginner	□ 30 mins	□ Student Quiz	□ Student Challenge

Prepare by familiarising yourself with internet components like devices, servers, ISPs, and routers. Use a relatable scenario of sending a message to explain the journey from sender to receiver. Engage students by letting them choose their message type. Highlight the roles of each internet component in the message's travel. Conclude with a summary of the process, emphasising packet transmission and reassembly.

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 components and their roles in the Internet's operation. 2. Recognise how data travels from sender to receiver across the Internet. 3. Appreciate the process of message transmission, including packetisation and routing. 4. Develop a conceptual model of Internet communication using a narrative approach. 5. Gain insight into the collaborative nature of Internet technology.	1. Identify and describe the roles of key Internet components (Sender, Receiver, Post Offices, Delivery Team, Traffic Managers) in message transmission. 2. Explain the process of sending a message from a Sender to a Receiver via the Internet. 3. Describe how messages are broken into packets and reassembled at the destination. 4. Illustrate the journey of a message through servers and routers using a narrative. 5. Summarise the overall function of the Internet in connecting devices for communication.

Lesson: Different Types of Devices

□ Beginner	□ 30 mins	□ Student Quiz	□ Student Challenge

Lesson Title: Different Types of Devices Teaching Notes: 1. Prepare a presentation covering the different types of devices: computers, smartphones, gaming consoles, smart TVs, smartwatches, e-readers, and smart home devices. Include images and key features of each device. 2. Be ready to explain how each device connects to the internet and its uses. 3. Prepare questions to engage students during the lesson, such as asking them to name devices they use at home or school. 4. For the concluding activity, ensure students understand the task. If possible, provide a worksheet for them to record their findings. 5. Encourage students to share their findings and discuss the importance of these devices in their daily lives.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• Understand the basic functions and characteristics of various digital devices such as computers, smartphones, gaming consoles, smart TVs, smartwatches, e-readers, and smart home devices. • Recognize the significance of internet connectivity in these devices and how it enhances their functionality. • Identify the differences and similarities between these devices in terms of their use, portability, and connectivity. • Appreciate the role of these devices in our daily lives, from communication and entertainment to information access and home automation. • Develop the ability to critically analyze the suitability of different devices for different tasks or needs. • Apply the knowledge gained by identifying and counting the different devices in their own environment.	• Identify and describe the functions of different types of devices such as computers, smartphones, gaming consoles, smart TVs, smartwatches, e-readers, and smart home devices. • Explain how each type of device connects to the internet and why this connection is important. • Differentiate between the various types of devices based on their uses, features, and connectivity options. • Recognize popular brands and models within each type of device category. • Apply knowledge of devices to identify and count the number of different devices in a given environment. • Communicate findings and share knowledge about different types of devices with peers or family members.

Lesson: What is Personal Information?

□ Beginner	□ 30 mins	□ Student Quiz	□ Student Challenge

This lesson aims to educate students about the concept of personal information, its importance, and the potential risks associated with sharing it online. Teachers should prepare examples of personal information and discuss why it's crucial to keep such information private. The lesson should also cover the dangers of sharing personal information on the internet and provide practical tips for protecting personal information. Teachers should encourage students to reflect on their online habits and consider changes to enhance their online safety. Reinforce the importance of privacy and caution when interacting online.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• Understand the definition of personal information and why it is important. • Identify examples of personal information. • Understand the risks associated with sharing personal information online. • Learn strategies for protecting personal information on the internet. • Reflect on personal online habits and identify areas for improvement in protecting personal information. • Develop an attitude of caution and responsibility when it comes to sharing personal information online.	• Students will be able to define what personal information is and provide examples of it. • Students will understand the importance of personal information and the potential risks associated with sharing it. • Students will be able to explain the relationship between personal information and internet safety. • Students will learn and be able to list strategies for protecting their personal information online. • Students will be able to assess their own online habits and identify areas where they can improve their personal information security. • Students will understand the importance of keeping their personal information private as a part of staying safe online.

Lesson: Why We Shouldn't Share Personal Information Online

□ Beginner	□ 30 mins	□ Student Quiz	□ Student Challenge

This lesson aims to educate students on the importance of online privacy and the risks associated with sharing personal information on the internet. Teachers should prepare real-life examples of each risk (identity theft, cyberbullying, unwanted contact, phishing scams) to help students understand the implications. Encourage students to share their experiences and thoughts on each topic. For younger students, consider role-playing scenarios to demonstrate how to handle such situations. The lesson should conclude with a discussion on the importance of reporting any online harassment or suspicious activity to a trusted adult.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• Understand what constitutes personal information and the potential risks associated with sharing it online. • Identify and comprehend the various risks of sharing personal information online, including identity theft, cyberbullying, unwanted contact from strangers, and phishing scams. • Recognize the implications of identity theft and how sharing personal information online can lead to such a situation. • Understand the concept of cyberbullying and how sharing personal information online can make one more vulnerable to it. • Learn how to avoid unwanted contact from strangers by being cautious about sharing personal information online. • Understand what phishing scams are and how they can be avoided by not sharing personal information online.	• Students will be able to define what constitutes personal information and understand the risks associated with sharing it online. • Students will be able to explain the concept of identity theft and how it can be facilitated by sharing personal information online. • Students will be able to describe the phenomenon of cyberbullying and understand how sharing personal information can make one vulnerable to it. • Students will be able to identify the dangers of unwanted contact from strangers online and how it can be avoided by not sharing personal information. • Students will be able to recognize phishing scams and understand how they can be used to trick individuals into sharing personal information. • Students will be able to apply the knowledge gained to protect their personal information and maintain online safety.

Lesson: Strong vs. Weak Passwords

□ Beginner	□ 30 mins	□ Student Quiz	□ Student Challenge

In this lesson, teachers will introduce the concept of passwords and their importance in online security. They will explain the difference between weak and strong passwords, using examples to illustrate. Teachers will then delve into the role of password length and character variety in enhancing password strength, using a table to demonstrate the time it would take to guess different types of passwords. The lesson concludes with a recap and an activity where students create their own strong passwords. Teachers should ensure students understand the importance of password security and the elements that constitute a strong password.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• Understand the importance of using strong passwords for online security. • Identify the characteristics of weak passwords and why they are easy to guess. • Understand the components of a strong password and how to create one. • Comprehend how the length and complexity of a password affects its security. • Analyze the strength of different types of passwords based on their composition and length. • Apply the learned knowledge to create and use strong, secure passwords in real-world contexts.	• Understand the importance of strong passwords and the risks associated with weak passwords. • Identify examples of weak passwords and explain why they are easy to guess or figure out. • Identify the characteristics of strong passwords and explain how they increase account security. • Understand how the length and complexity of a password affect its strength and the likelihood of it being hacked. • Analyze and compare the strength of different types of passwords based on their length and the types of characters used. • Create a strong password using the guidelines provided in the lesson.

Lesson: The Dangers of Sharing Passwords

□ Beginner	□ 30 mins	□ Student Quiz	□ Student Challenge

In this lesson, students will learn about the importance of password security. Teachers should emphasize the comparison of passwords to house keys, reinforcing the idea of privacy and security. The lesson will discuss the potential risks of password sharing, including identity theft and account misuse. Teachers should encourage students to never share passwords, even with friends, and to report any pressure to do so. The lesson concludes with practical tips for password protection. Teachers may want to role-play scenarios where students have to refuse password requests. Reinforce the importance of seeking adult guidance in uncertain situations.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• Understand the purpose and importance of passwords in protecting online accounts and personal information. • Recognize the potential dangers and consequences of sharing passwords. • Identify situations and individuals with whom passwords should never be shared. • Know how to respond if someone asks for their password. • Learn strategies for protecting and maintaining the security of their passwords. • Develop an attitude of responsibility and caution when it comes to handling their personal online security.	• Students will be able to explain the importance and purpose of passwords in protecting personal information and online accounts. • Students will be able to identify the potential dangers and consequences of sharing passwords. • Students will be able to list the individuals with whom they should never share their passwords. • Students will be able to describe appropriate actions to take if someone asks for their password. • Students will be able to demonstrate understanding of how to protect their passwords effectively. • Students will be able to understand the importance of seeking guidance from a trusted adult when faced with a situation involving password security.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Coding Short Course / Module: Intermediate Coding

Intermediate Coding

Lesson: Introduction to HTML

□ Beginner	□ 35 mins	□ Student Quiz

Prepare to introduce HTML as the standard markup language for creating web pages. Explain the structure of a web page. Discuss HTML elements, their start tags, content, and end tags. Highlight how web browsers interpret HTML code to display web pages. Facilitate hands-on practice with writing basic HTML code and adding content. Encourage students to experiment with their own details.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the purpose and structure of HTML in web development. 2. Identify and describe the key components of a web page structure. 3. Recognise and use basic HTML elements in coding. 4. Explain the role of web browsers in interpreting and displaying HTML code. 5. Apply knowledge to write and modify basic HTML code to create a simple web page.	1. Define HTML and its role in web page creation. 2. Identify and explain the structure of a web page using HTML. 3. Recognise and describe HTML elements and their functions. 4. Understand how web browsers interpret and display HTML code. 5. Write and run basic HTML code to create a simple web page.

Lesson: HTML Basic Elements

□ Beginner	□ 30 mins	□ Student Quiz

Prepare to guide students through understanding basic HTML elements. Start with explaining heading tags, their importance and usage. Move on to defining paragraphs, line breaks, and how to incorporate images with attributes. Discuss the concept of links, their attributes and how to create them. Finally, encourage students to write their own HTML code using these elements. Ensure to provide real-time examples and encourage hands-on practice.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand and apply HTML heading tags from h1 to h6. 2. Create and format paragraphs using the p tag. 3. Implement line breaks in HTML text using the br tag. 4. Insert and manipulate images using the img tag and its attributes. 5. Create hyperlinks to other web pages using the a tag and its attributes.	1. Identify and use HTML heading tags from <h1> to <h6>. 2. Define and implement paragraphs using the <p> tag. 3. Insert line breaks in HTML documents using the <br> tag. 4. Embed images in HTML using the <img> tag and its attributes. 5. Create hyperlinks using the <a> tag and understand the use of its attributes.

Lesson: HTML Tables

□ Beginner	□ 30 mins	□ Student Quiz

Prepare to guide students through the process of creating HTML tables. Begin with an introduction to tables and their structure, including headers, bodies, and footers. Then, delve into the specific HTML tags used to create tables, rows, and cells. Provide examples and encourage students to practice coding their own tables. Finally, challenge students to add multiple rows to their tables. Ensure students understand the importance of correctly nesting tags within each other.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the purpose and structure of HTML tables. 2. Identify and use HTML tags to create table headers, bodies, and footers. 3. Create table rows and columns using appropriate HTML tags. 4. Apply HTML coding to create a personalised table with multiple rows and columns. 5. Develop skills to troubleshoot and correct HTML table coding errors.	1. Identify and explain the structure and purpose of HTML tables. 2. Create a basic HTML table using the <table>, <tr>, and <td> tags. 3. Use <thead>, <tbody>, and <tfoot> tags to define the header, body, and footer of a table. 4. Develop a multi-row HTML table with specific content in each cell. 5. Apply CSS styling to HTML tables and their cells.

Lesson: Crafting Complex Tables

□ Intermediate	□ 30 mins	□ Student Quiz

Prepare to guide students through the process of crafting complex HTML tables. Ensure they understand basic table creation before introducing headers and footers. Highlight the importance of the 'rowspan' and 'colspan' attributes for merging cells. Encourage experimentation with these attributes to see their effects. Finally, review the completed table to ensure understanding.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Master the creation of basic HTML tables with multiple rows and columns. 2. Understand and apply the concept of table headers for better data context. 3. Learn to use the 'rowspan' attribute to merge cells vertically. 4. Learn to use the 'colspan' attribute to merge cells horizontally. 5. Develop the ability to add footer rows to tables for summary information.	1. Construct a basic HTML table with multiple rows and columns. 2. Integrate a header row into the table for column labelling. 3. Apply the 'rowspan' attribute to merge cells vertically. 4. Utilise the 'colspan' attribute to merge cells horizontally. 5. Add a footer row to the table for summary information.

Lesson: HTML Lists

□ Intermediate	□ 30 mins	□ Student Quiz

Prepare to introduce HTML lists, distinguishing between ordered and unordered types. Explain the use of `<ol></ol>` and `<li></li>` tags for ordered lists, and the `<ul></ul>` and `<li></li>` tags for unordered lists. Discuss the 'type' attribute for customising list markers. Demonstrate nested lists and encourage students to code their own lists, experimenting with different types and nesting.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the purpose and usage of HTML lists. 2. Code ordered and unordered lists in HTML. 3. Manipulate the numbering of ordered lists and bullet points of unordered lists. 4. Create nested lists in HTML. 5. Apply learned skills to code a personal list.	1. Identify and differentiate between ordered and unordered HTML lists. 2. Code ordered lists using the <ol> and <li> tags. 3. Manipulate the numbering type of ordered lists using the 'type' attribute. 4. Code unordered lists using the <ul> and <li> tags. 5. Alter the bullet point type of unordered lists using the 'type' attribute. 6. Create nested lists within both ordered and unordered lists. 7. Apply learned skills to code a personalised list.

Lesson: Basics of Form Creation

□ Intermediate	□ 60 mins	□ Student Quiz

In this lesson, teachers will guide students through the process of creating a basic HTML form. Starting with an introduction to HTML forms, students will learn how to create a form container, add input fields, labels, a textarea, and a submit button. The lesson encourages experimentation and exploration, allowing students to modify the code and observe the changes. Teachers should emphasise the importance of each element and attribute in the form, and how they contribute to the overall functionality and accessibility of the form.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the purpose and structure of HTML forms 2. Create a form container using the `<form>` element 3. Add and configure input fields for user data collection 4. Implement labels for enhanced accessibility and usability 5. Include a textarea for multi-line text input 6. Add a submit button to enable form submission	1. Understand and apply the HTML `<form>` element to create a form container. 2. Create and utilise `<input>` fields for text and email data collection. 3. Implement `<label>` elements for improved form accessibility and usability. 4. Add a `<textarea>` element for multi-line text input. 5. Include a `<button>` with a `type` attribute set to `submit` to finalise form creation.

Lesson: Advanced Input Types

□ Advanced	□ 60 mins	□ Student Quiz

This lesson explores advanced HTML form input types: number, date, and colour. Teachers should guide students through the creation of number, date, and colour input fields, demonstrating the enhanced functionality and user experience these types offer. The lesson concludes with students enhancing a form with these advanced input types, applying their newly acquired knowledge.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand and utilise advanced HTML input types including number, date, and colour. 2. Create a number input field for capturing numerical data such as age. 3. Implement a date input field for selecting specific dates. 4. Use a colour input field for selecting a colour from a colour picker. 5. Enhance a form by integrating advanced input types to improve functionality and user experience.	1. Understand and apply the number input type in HTML forms. 2. Understand and apply the date input type in HTML forms. 3. Understand and apply the color input type in HTML forms. 4. Create HTML forms utilising advanced input types. 5. Enhance user experience by implementing advanced input types in forms.

Lesson: Embedding Audio and Video

□ Advanced	□ 30 mins	□ Student Quiz

In this lesson, teachers will guide students through the process of embedding audio and video into a webpage using HTML5. They will start by discussing the importance of understanding supported formats for different web browsers. Students will then learn how to embed audio and video files using the HTML5 `<audio>` and `<video>` tags. Teachers will encourage students to test their work and reflect on their learning. The lesson will conclude with a review and encouragement for continued practice.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Identify and understand the different audio and video formats supported by HTML5. 2. Embed an audio file into a web page using the HTML5 <audio> tag. 3. Embed a video file into a web page using the HTML5 <video> tag. 4. Preview and test the functionality of embedded audio and video players. 5. Reflect on the process and importance of embedding multimedia elements in web pages.	1. Identify and understand the common audio and video formats supported by HTML5 and their compatibility with various web browsers. 2. Embed an audio file into a web page using the HTML5 <audio> tag and provide an MP3 file as the source. 3. Embed a video file into a web page using the HTML5 <video> tag and provide an MP4 file as the source. 4. Preview and test the functionality of the embedded audio and video players, and experiment with different video dimensions. 5. Reflect on the process of embedding audio and video files into a web page using HTML5 tags and understand the importance of compatibility and user experience.

Lesson: Introduction to CSS

□ Advanced	□ 30 mins	□ Student Quiz

Prepare to introduce students to CSS, the language for styling web pages. Explain what CSS is and how it interacts with HTML. Discuss CSS rules, selectors, and declaration blocks. Use examples to illustrate how CSS changes the appearance of HTML elements. Introduce the concept of element, ID, and class selectors. Explain the style property and how it can be used to directly apply CSS to HTML elements. Provide exercises for students to practice writing CSS code and applying different selectors.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the purpose and function of CSS in web development. 2. Identify and apply CSS rules including selectors and declaration blocks. 3. Differentiate between element, ID, and class selectors in CSS. 4. Apply CSS styles directly to HTML elements using the style property. 5. Practice creating and applying CSS classes to HTML elements.	1. Understand the purpose and function of CSS in web development. 2. Identify and apply CSS rules including selectors and declaration blocks. 3. Use CSS to style HTML elements using element, ID, and class selectors. 4. Apply CSS properties directly to HTML elements using the style property. 5. Practice writing CSS code through exercises and understand how it affects the appearance of HTML elements.

Lesson: CSS Box Model

□ Advanced	□ 30 mins	□ Student Quiz

Prepare to guide students through understanding the CSS Box Model, including margins, borders, padding, and content. Demonstrate how to apply different border styles, widths, and colours. Encourage students to experiment with padding and margins to understand their impact on layout. The lesson includes practical exercises to reinforce learning. Ensure students understand how to use the code examples provided.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the CSS Box Model and its components: margins, borders, padding, and content. 2. Apply different styles, widths, and colours to CSS borders. 3. Manipulate padding in CSS to create space around content within the border. 4. Use CSS margins to create space around elements, outside the border. 5. Perform exercises to apply CSS Box Model properties to HTML elements.	1. Identify and explain the components of the CSS Box Model. 2. Apply CSS properties to create and modify borders on HTML elements. 3. Use CSS properties to set border styles, widths, and colours. 4. Apply CSS padding to create space around content within an element. 5. Use CSS margins to create space around HTML elements.

Lesson: CSS Text

□ Advanced	□ 40 mins	□ Student Quiz

Prepare to guide students through the process of styling text using CSS. The lesson covers setting text and background colours, alignment, decoration, transformation, spacing, and adding a shadow. Students will learn to use properties such as 'color', 'text-align', 'text-decoration', 'text-transform', 'letter-spacing', 'word-spacing', 'line-height', 'text-indent', and 'text-shadow'. They will also experiment with different values for these properties, including colour names, HEX values, RGB values, and pixel sizes.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand and apply CSS properties to style text colour and background colour. 2. Manipulate text alignment using CSS. 3. Use CSS to add or remove text decorations. 4. Transform text to uppercase, lowercase or capitalised format using CSS. 5. Apply CSS properties to adjust text spacing and add text shadow.	1. Apply CSS properties to style text colour and background colour. 2. Align text using CSS properties. 3. Decorate text using underline, overline and line-through CSS properties. 4. Transform text to uppercase, lowercase and capitalise using CSS properties. 5. Apply CSS properties to set text spacing and add text shadow.

Lesson: CSS Fonts

□ Advanced	□ 40 mins	□ Student Quiz

Prepare to introduce students to CSS fonts, explaining their importance in web design. Discuss the 'font-family' property and provide examples of commonly used font families. Explain the concept of 'web safe fonts' and the use of fallback fonts. Introduce 'font-weight', 'font-size', and 'font-style' properties. Prepare an exercise where students will code a paragraph of text using the discussed properties. Be ready to provide a solution and explain the code.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the importance of font selection in CSS and how it impacts user experience. 2. Apply the 'font-family' property to set specific fonts for text elements. 3. Use 'font-family' to specify fallback fonts when the primary font is unavailable. 4. Manipulate 'font-weight', 'font-size', and 'font-style' properties to modify the appearance of text. 5. Implement learned CSS font properties in a practical exercise.	1. Apply the CSS property 'font-family' to set specific fonts for text. 2. Specify 'fallback' fonts using the 'font-family' CSS property. 3. Manipulate the weight of the font using the 'font-weight' CSS property. 4. Adjust the size of the font using the 'font-size' CSS property. 5. Change the style of the font using the 'font-style' CSS property.

Lesson: CSS Website Layout

□ Advanced	□ 40 mins	□ Student Quiz

This lesson focuses on CSS website layout. Teachers should familiarise themselves with the basic structure of a website, including the header, content, and footer. The lesson covers how to code these areas using HTML and CSS, with practical examples provided. It also explores different content layouts, such as one-column, two-column, and three-column layouts. The lesson concludes with a comprehensive example of putting all the elements together to create a complete website layout. Teachers should encourage students to experiment with the code examples provided.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the structure of a website layout including header, content, and footer. 2. Develop skills to code and style a website header using HTML and CSS. 3. Learn to create different content layouts such as one-column, two-column, and three-column layouts. 4. Gain proficiency in using CSS to set column widths and layout. 5. Acquire knowledge to code and style a website footer using HTML and CSS.	1. Identify and code common website layout areas: header, content, and footer using CSS. 2. Construct and style a website header with logo and navigation menu. 3. Design and implement one, two, and three column content layouts. 4. Manipulate column widths and padding to achieve desired layout. 5. Create and style a website footer with company information and secondary links. 6. Combine all elements to create a cohesive website layout.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Coding Short Course / Module: Intermediate Coding

Intermediate Coding

Lesson: Microbit Light Clapper

□ Beginner	□ 40 mins	□ Student Quiz	□ Student Challenge

Prepare for the 'Microbit Light Clapper' lesson by familiarising yourself with the makecode.microbit.org website. Understand the process of creating a new project, setting up variables, and using sound thresholds. Be ready to guide students in writing code to detect claps and control LED lights. Ensure you can troubleshoot issues and explain how to test the code in the simulator and on the Microbit.

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 concept of variables in coding. 2. Set and utilise sound thresholds for input detection. 3. Implement conditional statements (if-then-else) to control LED light responses. 4. Test and debug code in a simulator environment. 5. Transfer and apply code to a physical Microbit device.	1. Develop a new project using makecode.microbit.org. 2. Create and utilise a variable to control the LED lights on the Microbit. 3. Set a sound threshold for detecting claps using the Microbit's microphone. 4. Implement code to detect a clap based on the set sound threshold. 5. Use an 'if then else' block to control the LED lights based on the clap detection.

Lesson: Microbit Sounds

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare by ensuring access to MakeCode and micro:bit devices. Begin with an introduction to sound programming. Guide students through creating a new project named 'Music Maker'. Demonstrate playing melodies with the 'A' button, tones with the 'B' button, and sound effects with 'A+B'. Encourage experimentation with different musical elements. Conclude by testing the music on simulators or physical devices.

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 basics of programming music and sound on a microbit. 2. Create and modify melodies using different notes and tempos. 3. Program and experiment with playing individual tones. 4. Design and adjust sound effects using various waveforms and settings. 5. Test and refine music and sound projects on a microbit.	1. Create a new microbit project named 'Music Maker'. 2. Program the 'A' button to play a custom melody using the 'music.play' function. 3. Program the 'B' button to play a sequence of tones using the 'music.playTone' function. 4. Program the A+B buttons to play a custom sound effect using the 'music.playSoundEffect' function. 5. Test the music by pressing buttons on the simulator or a physical Microbit.

Lesson: Higher or Lower Game

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare for this lesson by familiarising yourself with the Microbit project creation process and the coding involved in creating variables. Understand the game setup, including the use of random numbers and how they're displayed. Be prepared to explain the game mechanics, such as guessing higher or lower, scoring points, and resetting numbers for the next round. Be ready to guide students through the game over process and the steps to duplicate code for the 'higher' guess. Finally, ensure you know how to download the game onto a Microbit for playing.

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 in creating and manipulating variables within a Microbit project. 2. Understand and apply the concept of random number generation in game development. 3. Gain proficiency in programming button inputs to trigger specific actions. 4. Learn to implement scoring systems and game over conditions in a game project. 5. Enhance problem-solving skills by debugging and testing a game on a Microbit device.	1. Create a new Microbit project and two variables for the game. 2. Set up the start of the game with random numbers and display the number on the Microbit. 3. Program the A button to guess if the next number is lower and score a point if the guess is correct. 4. Reset the variables for the next round after a correct guess and display the new number on the Microbit. 5. End the game if the guess is incorrect and display the final score. 6. Program the B button to guess if the next number is higher, following the same rules as the A button. 7. Download the game onto the Microbit and play it.

Lesson: Microbit Paddle Ball

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

In this lesson, students will create a Microbit Paddle Ball game using Scratch. They will learn to create a new project, add and position sprites, and make the ball bounce around the screen. They will also connect a Microbit to control the paddle, make the ball bounce off the paddle, add a backdrop, and create a game over line. The lesson concludes with programming the game over functionality and discussing potential improvements to the game. Teachers should familiarise themselves with Scratch and Microbit prior to the lesson.

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 in creating and managing a new Scratch project. 2. Understand and apply the concept of adding and positioning sprites in Scratch. 3. Gain proficiency in coding for sprite movement, interaction, and control using Scratch blocks. 4. Learn to integrate and use a Microbit with Scratch for real-time control of sprites. 5. Enhance critical thinking and problem-solving skills by identifying potential improvements to the game.	1. Develop a new Scratch project and remove the default cat sprite. 2. Add and position the 'Paddle' sprite from the sprite library. 3. Add the 'Soccer Ball' sprite from the sprite library. 4. Set the X and Y coordinates to position the ball at the top center of the screen. 5. Code the ball to move around the screen and bounce off the edges. 6. Connect and configure a Microbit to the Scratch project. 7. Code the paddle to move left and right by tilting the Microbit. 8. Program the ball to bounce off the paddle when it touches it. 9. Add the 'Stars' backdrop from the backdrop library. 10. Draw a red line at the bottom of the screen for the game over line. 11. Code the game to end when the ball touches the red game over line. 12. Propose improvements to the game by adding to or changing the existing code.

Lesson: Microbit Seismic and Meteorological Station

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

In this step-by-step lesson, you'll programme four Microbits to create a Seismic and Meteorological Station. Each Microbit will monitor and display different data: temperature, light levels, and simulated seismic activity. You'll then test your station, and explore ways to enhance it, such as recording historical data or setting specific alert conditions.

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 role of each Microbit in the Seismic and Meteorological Station and how they communicate with each other. 2. Develop skills in coding and programming Microbits to monitor and display temperature, light levels, and seismic activity. 3. Test and troubleshoot the coded Microbits to ensure they function as intended in the Seismic and Meteorological Station. 4. Apply critical thinking to improve the functionality of the Seismic and Meteorological Station. 5. Develop an understanding of how meteorological and seismic data can be collected, displayed, and used in real-world applications.	1. Program four separate Microbits to perform unique roles in a Seismic and Meteorological Station. 2. Code the 'Seismic and Meteorological Station' Microbit to monitor and wirelessly broadcast temperature, light levels, and 'seismic' activity data. 3. Code the 'Temperature Display' Microbit to receive and display the temperature data from the 'Seismic and Meteorological Station' Microbit. 4. Code the 'Day/Night Indicator' Microbit to receive light level data and display an indication of whether it's day or night. 5. Code the 'Seismic Alert' Microbit to receive 'seismic' activity data and display an alert if seismic activity is detected.

Lesson: Microbit Radio Messages

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Ensure you have two Microbits and understand the concept of a radio transceiver. Familiarise yourself with the MakeCode website and how to create a new project. Understand the importance of setting a radio group for communication. Be prepared to guide students on how to send different types of messages (string, number, name and value) and how to receive and display these messages. Encourage exploration of the Radio blocks for creative coding.

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 radio transceiver feature of Microbits for communication. 2. Create a new project using the MakeCode Microbit website. 3. Set and comprehend the function of radio groups in Microbit communication. 4. Send, receive, and display different types of messages (string, number, name-value) using Microbits. 5. Explore and experiment with the Radio blocks for various coding possibilities.	1. Understand and utilise the radio transceiver feature of Microbits for communication. 2. Create a new project on the Microbit website and set the radio group for broadcasting. 3. Send and receive different types of messages (string, number, name and value) using Microbits. 4. Apply the code to both Microbits and test the functionality of the radio messages. 5. Explore and experiment with the Radio blocks for various coding possibilities.

Lesson: Microbit Voting System

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students in creating a microbit voting system. They'll create two projects on the MakeCode Microbit website, one for voting microbits and another for a central microbit. They'll program the A and B buttons to cast votes, set up the central microbit to receive votes and reset the system, and display the vote results. They'll also enhance the system with a security feature. Ensure students understand the coding involved and the importance of testing their system.

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 a microbit voting system with separate voting and central microbits. 2. Programme the voting microbits to cast a single 'Yes' or 'No' vote. 3. Configure the central microbit to receive votes, count them, and reset the voting system. 4. Implement a reset function on the voting microbits to allow for multiple rounds of voting. 5. Enhance the voting system by adding a security feature to ensure the integrity of the votes.	1. Develop two separate projects on the MakeCode Microbit website for voting and central microbits. 2. Programme the A and B buttons on the microbit to cast a single 'Yes' or 'No' vote. 3. Set up the central microbit to receive votes, count them, and reset the voting system when needed. 4. Configure the individual voting microbits to receive the 'Reset' signal from the central microbit. 5. Display the vote results on the central microbit.

Intermediate Coding

Lesson: An Introduction to AI Models

□ Advanced	□ 20 mins	□ Student Quiz

Prepare to introduce students to AI models, explaining their function and various types. Discuss different learning methods such as supervised, unsupervised, and reinforcement learning. Explore the diverse applications of AI models, from speech recognition to autonomous vehicles. Discuss the limitations of AI models, including data quality and computational resources. Finally, delve into the ethics of AI models, discussing responsibility, privacy, transparency, and fairness.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the concept and purpose of AI models. 2. Identify different types of AI models and their learning methods. 3. Recognise various applications of AI models in real-world scenarios. 4. Appreciate the limitations and challenges associated with AI models. 5. Reflect on the ethical considerations in the use of AI models.	1. Identify and describe the different types of AI models: Supervised Learning, Unsupervised Learning, and Reinforcement Learning. 2. Explain the various applications of AI models, including speech recognition, image recognition, natural language processing, recommendation systems, and autonomous vehicles. 3. Discuss the limitations of AI models, focusing on data quality, computational resources, transparency, privacy, and security. 4. Understand the ethical considerations related to AI models, including responsibility, privacy, transparency, and fairness. 5. Demonstrate a basic understanding of how AI models function, their uses, limitations, and ethical implications.

Lesson: Create an Image Model

□ Advanced	□ 20 mins	□ Student Quiz

Familiarise yourself with Google's Teachable Machine tool before the lesson. Ensure students understand the concept of machine learning and how it applies to image recognition. Encourage students to take clear images for their classes and emphasise the importance of quality over quantity. Guide them through the process of training, testing, and exporting their models. Reinforce the practical application of these skills in future projects.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Required equipment for this lesson:

- Webcam/camera

Learning Goals	Learning Outcomes
1. Understand and utilise Google's Teachable Machine to create an image model. 2. Create and define classes within an image model project. 3. Add and manage image samples to each class for effective model training. 4. Train, test, and refine the image model to ensure accurate gesture recognition. 5. Export and save the created image model for future use in projects.	1. Utilise Google's Teachable Machine to create an image model. 2. Create and categorise classes within an image model project. 3. Add and record images to each class using a webcam. 4. Train the image model using the added images and understand the process of machine learning. 5. Test the model's performance, make necessary adjustments, and export the model for future use.

Lesson: Scratch AI Rock, Paper, Scissors Game

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through creating a Rock, Paper, Scissors game using Scratch and Google Teachable Machine. Ensure they understand the use of variables, randomisation, and conditionals. They'll need to set up Scratch and TM2Scratch, add a sprite, create variables, and load a Teachable Machine Model. They'll also learn to set a confidence threshold, get player choice, determine game outcomes, and add enhancements. Encourage creativity and problem-solving throughout.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Required equipment for this lesson:

- Webcam/camera

Learning Goals	Learning Outcomes
1. Develop a Rock, Paper, Scissors game using Scratch and Google Teachable Machine. 2. Understand and apply the use of variables in Scratch for storing player's choice, computer's choice, and the result of the game. 3. Implement randomisation in Scratch to simulate the computer's choice in the game. 4. Integrate Google Teachable Machine Image models in Scratch projects for gesture recognition. 5. Understand and adjust the confidence threshold for AI model to improve accuracy of gesture recognition.	1. Develop a Rock, Paper, Scissors game using Scratch and Google Teachable Machine. 2. Set up Scratch and TM2Scratch for the game development. 3. Create and utilise variables to store player's choice, computer's choice, and the game result. 4. Implement randomisation for computer's choice in the game. 5. Load and use a Teachable Machine Image model for hand gesture recognition. 6. Set and adjust the confidence threshold for the AI model. 7. Recognise and interpret player's choice through hand gestures. 8. Develop game logic to determine the game result: draw, win, or lose. 9. Improve the game by enhancing the image model, adding new features like sound effects, and improving user interaction.

Lesson: Create a Pose Model

□ Advanced	□ 20 mins	□ Student Quiz

Prepare to guide students through creating a pose model using Google's Teachable Machine. Familiarise yourself with the tool and the process of creating classes, adding images, and training the model. Be ready to troubleshoot any issues with webcam permissions or image quality. Ensure students understand the importance of testing their model and making necessary adjustments. Finally, assist them in exporting their model for future use in projects like an AI-powered space game.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Required equipment for this lesson:

- Webcam/camera

Learning Goals	Learning Outcomes
1. Develop an understanding of Google's Teachable Machine and its application in creating pose models. 2. Acquire skills to create and categorise classes within a pose model. 3. Learn to add and manage image samples for each class to train the model. 4. Gain proficiency in training and testing the model for different poses. 5. Master the process of exporting the model for future use in other projects.	1. Operate Google's Teachable Machine to create a pose model. 2. Define and create classes for the pose model. 3. Add and categorise images into the respective classes: Tilt Left, Tilt Right, and No Tilt. 4. Train the pose model using the categorised images and test its performance. 5. Export the created pose model and obtain a shareable link for future use.

Lesson: Scratch AI Pose Space Game

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students in creating a Scratch AI Pose Space Game. They'll learn to use Scratch and Google Teachable Machine to control a spaceship with tilt poses. They'll set up Scratch, add a rocketship sprite, load a Teachable Machine Model, display pose labels, set a confidence threshold, and make the spaceship move. They'll also add a star sprite, make stars fall, and face a challenge to improve their game. Ensure students understand each step, and encourage creativity in the challenge.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Required equipment for this lesson:

- Webcam/camera

Learning Goals	Learning Outcomes
1. Develop skills in using Scratch and Google Teachable Machine to create a game. 2. Understand how to control a sprite using pose models. 3. Learn to set up and adjust the confidence threshold for an AI model. 4. Gain knowledge on how to create and manipulate clones of sprites in Scratch. 5. Apply creativity to enhance and personalise the game with additional features.	1. Create a Space game using Scratch and Google Teachable Machine. 2. Set up Scratch and TMPose2Scratch for a new project. 3. Integrate a Teachable Machine Pose model into the Scratch project. 4. Control a sprite's movement using pose labels and confidence thresholds. 5. Enhance the game by adding falling sprites and scoring mechanisms.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Coding Short Course / Module: Advanced Coding

Advanced Coding

Lesson: Microbit Sensor Graphs

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

For the 'Microbit Sensor Graphs' lesson, teachers should familiarise themselves with the makecode.microbit.org website and how to create a new project. They should understand how the light level sensor works and how to display and graph the light level. Teachers should also explore how to graph other sensors such as temperature, compass heading, acceleration, magnetic forces, and sound level. Lastly, they should encourage students to get creative with their graphs and consider how these sensors could be used in other projects.

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 how to create a new project on makecode.microbit.org. 2. Learn to display the light level on the Microbit. 3. Develop skills to graph the light level and observe changes. 4. Gain knowledge on graphing other sensors such as temperature, compass heading, acceleration, magnetic forces and sound level. 5. Apply creativity to graph other possible values and incorporate sensor usage in different projects.	1. Create a new project on makecode.microbit.org. 2. Display the light level on the Microbit. 3. Graph the light level changes on the Microbit. 4. Graph the readings from other sensors on the Microbit, including temperature, compass heading, acceleration, magnetic forces, and sound level. 5. Design and implement a creative application using the Microbit's sensors.

Lesson: Creating a Microbits Alarm System

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare for the 'Creating a Microbits Alarm System' lesson by familiarising yourself with the Microbits MakeCode editor. Understand the process of creating new projects and setting up variables. Grasp the concept of arming and disarming the alarm system using button inputs. Understand how to define a function for the alarm and set up triggers based on sound and light thresholds. Finally, be ready to guide students through testing their alarm systems in a simulator or on a physical 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 concept of variables in Microbits programming. 2. Develop skills to use input functions for button presses on the Microbit device. 3. Learn to create and utilise functions for specific tasks in coding. 4. Gain knowledge on using sensor inputs (sound and light) to trigger events. 5. Apply testing and debugging skills to ensure the functionality of the Microbits alarm system.	1. Develop a new project using the Microbits MakeCode editor. 2. Establish sound and light threshold variables for the alarm system. 3. Implement a function to arm the alarm system using the A button on the Microbit. 4. Design a function to disarm the alarm system using the B button on the Microbit. 5. Define a function to sound and flash the alarm when triggered. 6. Set up alarm triggers that monitor sensor values and activate the alarm when thresholds are crossed. 7. Test the alarm system in a simulator and on a physical Microbits device.

Lesson: Exactly 11

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

This lesson involves creating a Microbit project where students guess when 11 seconds have passed. They will learn to create and use variables 'starttime', 'taken', and 'difference'. They will also learn to use the 'running time (ms)' block, calculate absolute values, and use conditional statements to display results. The lesson involves coding and understanding the concept of milliseconds.

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 in creating and utilising variables in Microbit projects. 2. Understand and apply the concept of running time in programming. 3. Gain proficiency in using mathematical operations to calculate time differences. 4. Learn to use conditional statements to display different outcomes based on user input. 5. Enhance problem-solving skills through the creation of a time-guessing game.	1. Develop a new Microbit project. 2. Create and utilise 'starttime' variable to record the start time. 3. Establish a 'taken' variable to store the time elapsed between two actions. 4. Formulate a 'difference' variable to calculate the difference between 11 seconds and the 'taken' time. 5. Implement code to display the result and provide feedback to the user.

Lesson: Microbit Finder

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare two Microbits and ensure one is portable. The lesson involves creating a code to be downloaded onto both Microbits, using A and B buttons to set 'lost' and 'finder' Microbits. The project requires creating two variables, setting a radio group for communication, programming buttons to set modes, sending and receiving messages, and displaying proximity. The code is then downloaded onto both Microbits for a practical demonstration.

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 Microbit's radio communication feature. 2. Create and manipulate variables in a Microbit project. 3. Program Microbit's buttons to perform specific actions. 4. Interpret signal strength to determine proximity between two Microbits. 5. Apply coding skills to create a practical Microbit application.	1. Programme two Microbits to act as a 'finder' and a 'lost' device using A and B buttons. 2. Create and utilise 'mode' and 'signal' variables to control Microbit actions. 3. Set up a radio group for Microbits to communicate with each other. 4. Code the 'lost' Microbit to continuously send a signal for the 'finder' Microbit to detect. 5. Interpret the signal strength of the received message to determine the proximity of the 'lost' Microbit.

Lesson: Chase the Dot

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

In this lesson, students will create a game called 'Chase the Dot' using Microbit. They will learn to create a new project, define variables, create a function, and use gestures to control movements. The game involves two dots, a target and a chaser. The aim is for the chaser dot to catch the target dot, which moves to a random position each time it's caught. The lesson involves coding for dot creation, movement, scoring, and game initiation.

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 in creating and managing Microbit projects on makecode.com. 2. Understand and apply the concept of variables in coding to create game sprites. 3. Learn to create and use functions for repetitive tasks in a game scenario. 4. Develop skills in using gestures to control game elements in a Microbit project. 5. Understand and implement the concept of scoring and round systems in game development.	1. Create and manipulate variables to store and control game sprites in a Microbit project. 2. Develop a function to position a sprite at a random location on the edge of the screen. 3. Implement a countdown timer and sound effects to enhance game play. 4. Programme the Microbit to respond to tilt gestures to control sprite movement. 5. Design a scoring system that responds to sprite interactions and triggers new rounds of play.

Lesson: Microbit - Invaders

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through creating a Microbit project on the Microbit website. They'll learn to create a countdown timer, and establish variables for Bullets, Players, and Enemy. Students will then create and move player and enemy sprites, incorporating logic for movement and game-ending conditions. They'll also create a bullet sprite, with movement and enemy-hit detection. Lastly, they'll add conditions for enemy sprites hitting the player, ending the game.

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 a Microbit project using the online platform. 2. Create a countdown timer using basic show number and pause blocks. 3. Define and utilise variables for game elements such as Bullets, Players and Enemy. 4. Design and implement player and enemy sprites in the game. 5. Enable player sprite movement using A and B button inputs.	1. Create a Microbit project using the Microbit website. 2. Develop a countdown timer using the show number and pause blocks in the Basic category. 3. Create three variables: Bullets, Players, and Enemy. 4. Design a player sprite and position it on the left side of the screen. 5. Program the player sprite to move left or right in response to the A and B buttons. 6. Create an enemy sprite, position it at the top of the screen, and program it to move downwards and end the game if it touches the player sprite. 7. Create a bullet sprite that moves upwards from the player's position when button A+B is pressed. 8. Implement a condition to check if the bullet sprite is touching the enemy sprite, and if so, make the enemy sprite disappear and add a point to the player's score. 9. Add a condition to check if the enemy sprite is touching the player sprite, and if so, end the game.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Coding Short Course / Module: Advanced Coding

Advanced Coding

Lesson: Introduction to JavaScript

□ Intermediate	□ 60 mins	□ Student Quiz

This lesson introduces students to JavaScript, a popular programming language. Teachers should explain the concept of programming languages and how JavaScript is used to make websites interactive. The lesson progresses through a sequence of steps, starting with basic JavaScript code and gradually introducing more complex elements. Students will have the opportunity to write their own JavaScript code, switch between block and JavaScript views, and correct errors in their code. The lesson concludes with a challenge to program a button to play music. Teachers should be prepared to assist students and answer questions throughout the lesson.

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

- Chromebook/Laptop/PC

Learning Goals	Learning Outcomes
1. Understand the basic concept and importance of JavaScript in web development. 2. Learn about the sequence of execution in JavaScript programming. 3. Gain hands-on experience in writing and executing JavaScript code. 4. Develop skills to troubleshoot and correct syntax errors in JavaScript. 5. Apply JavaScript knowledge to create interactive functions, such as programming buttons and displaying messages.	1. Understand the basic concept and purpose of JavaScript as a textual programming language. 2. Recognise the sequence in which JavaScript code is executed. 3. Write and modify simple JavaScript code using the Makecode Microbit project editor. 4. Identify and correct common syntax errors in JavaScript code. 5. Program interactive functions in JavaScript, such as button presses and music playback.

Lesson: JavaScript - Exactly 11

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through creating a game using JavaScript, focusing on variables, functions, and mathematical operations. The game, 'Exactly 11', challenges players to guess when exactly 11 seconds have passed. Students will learn to create and use variables such as 'starttime', 'taken', and 'difference', and to use functions like 'input.onButtonPressed' and 'Math.abs'. They will also learn to add comments to their code for clarity. The lesson concludes with testing the game in the Microbit simulator.

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

- Chromebook/Laptop/PC

Learning Goals	Learning Outcomes
1. Develop understanding of variables, functions, and mathematical operations in JavaScript. 2. Gain proficiency in creating and manipulating variables such as 'starttime', 'taken', and 'difference'. 3. Learn to use 'input.runningTime()' to measure time in JavaScript. 4. Understand the use of 'Math.abs()' for calculating absolute values. 5. Apply conditional statements to display different outcomes based on user input.	1. Create a new Microbit project using JavaScript. 2. Define and initialise 'starttime' variable in JavaScript. 3. Implement a function to set the start time using 'input.onButtonPressed' method. 4. Create and initialise 'taken' and 'difference' variables in JavaScript. 5. Program a function to calculate the time taken and the difference from 11 seconds. 6. Display the result of the game using conditional statements and 'basic.showIcon' method.

Lesson: JavaScript Variables

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to introduce students to JavaScript variables, explaining their role as containers for data values. Use practical examples such as a game score to illustrate this concept. Guide students through the process of creating, adding to, and displaying variables. Progress to discussing number and string variables, demonstrating how they can be added or concatenated. Explain the importance of unique identifiers for variables and the rules for constructing them. Facilitate an exercise where students create their own variables, and guide them through programming buttons to display these variables. Finally, simulate a birthday scenario to demonstrate how variable values can be updated.

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

- Chromebook/Laptop/PC

Learning Goals	Learning Outcomes
1. Understand and apply the concept of variables in JavaScript. 2. Perform arithmetic operations with number variables. 3. Concatenate string variables to form new strings. 4. Identify and follow the rules for naming JavaScript variables. 5. Write JavaScript code to create and manipulate variables.	1. Define and initialise JavaScript variables. 2. Perform arithmetic operations with number variables. 3. Concatenate string variables. 4. Understand and apply rules for JavaScript identifiers. 5. Manipulate variables through button-press and gesture-based events in JavaScript code.

Lesson: JavaScript Data Types

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through understanding JavaScript data types, including strings, numbers, booleans, arrays, and objects. Emphasise the importance of correct syntax and the role of quotes in defining strings. Highlight the use of booleans in conditional statements. Explain how arrays store multiple values and how objects are collections of properties. Discuss the concept of 'undefined' and its implications. Finally, facilitate hands-on exercises to reinforce learning, using the makecode.microbit.org platform for practical application.

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

- Chromebook/Laptop/PC

Learning Goals	Learning Outcomes
1. Understand and apply JavaScript data types including strings, numbers, booleans, arrays, and objects. 2. Recognise and handle undefined variables in JavaScript. 3. Manipulate strings using single and double quotes. 4. Use booleans in conditional statements. 5. Create and manipulate arrays and objects, including accessing specific elements and properties.	1. Identify and differentiate between various JavaScript data types including strings, numbers, booleans, arrays, objects, and undefined. 2. Write and manipulate JavaScript strings using single and double quotes. 3. Understand and utilise JavaScript numbers with or without decimal points. 4. Implement JavaScript booleans in conditional statements. 5. Create and manipulate JavaScript arrays using square brackets and commas. 6. Construct JavaScript objects using curly braces, and understand the concept of name:value pairs. 7. Recognise and handle undefined variables in JavaScript. 8. Apply learned JavaScript data types in practical coding exercises. 9. Understand and use array indexing to access specific elements in a JavaScript array. 10. Manipulate array indexes using button-pressed functions in JavaScript.

Lesson: JavaScript Operators

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to introduce students to JavaScript operators, starting with the basics of what an operator is. Use practical examples to explain different types of operators including arithmetic, string, assignment, comparison, and logical operators. Encourage students to try out each operator type with hands-on coding exercises on the MakeCode website. Ensure students understand the concept of operands and how operators are used to perform operations on these operands.

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

- Chromebook/Laptop/PC

Learning Goals	Learning Outcomes
1. Understand and apply JavaScript operators including arithmetic, string, assignment, comparison, and logical operators. 2. Perform arithmetic operations using JavaScript arithmetic operators. 3. Manipulate strings using JavaScript string operators. 4. Assign and modify variable values using JavaScript assignment operators. 5. Compare values and make decisions using JavaScript comparison and logical operators.	1. Identify and explain the different types of JavaScript operators. 2. Perform arithmetic operations using JavaScript arithmetic operators. 3. Concatenate strings and perform operations on variables using JavaScript string and assignment operators. 4. Compare values using JavaScript comparison operators. 5. Apply logical operators to determine the logic between variables or values in JavaScript.

Lesson: JavaScript Conditional Statements

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

This lesson delves into JavaScript Conditional Statements, starting with an introduction to 'if', 'else if', and 'else' statements. Teachers should guide students through the process of writing and understanding these statements, using practical examples such as setting movie ticket prices based on age. The lesson then progresses to more complex scenarios involving multiple conditions. Students will get hands-on experience by writing their own conditional statements to display different greeting messages based on the time of day. Teachers should ensure students understand the importance of correct syntax and the use of operators in these statements.

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

- Chromebook/Laptop/PC

Learning Goals	Learning Outcomes
1. Understand and apply JavaScript conditional statements including 'if', 'else if', and 'else'. 2. Use conditional statements to test conditions and execute different code blocks based on the results. 3. Combine multiple conditions in a single 'if' statement using logical operators. 4. Write and implement custom 'if', 'else if', and 'else' statements in practical scenarios. 5. Debug and improve code using conditional statements to enhance program functionality.	1. Understand the concept and purpose of JavaScript conditional statements including 'if', 'else if', and 'else'. 2. Apply 'if' statement in JavaScript to test a condition and execute a piece of code if the condition is true. 3. Utilise 'else if' and 'else' statements in JavaScript to test multiple conditions and execute different pieces of code based on the results. 4. Write a JavaScript 'if' statement to set a variable based on a condition. 5. Develop a JavaScript program using 'if', 'else if', and 'else' statements to set a variable based on multiple conditions.

Lesson: JavaScript Switch Statements

□ Advanced	□ 25 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through the concept of JavaScript Switch Statements. Begin with explaining what a switch statement is, using the provided examples. Highlight the importance of the 'break' keyword and its role in the flow of code execution. Introduce the 'default' keyword and its use when none of the case conditions are true. Discuss how the same code can be run for different case conditions. Finally, encourage students to write their own switch statement and include a default case.

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

- Chromebook/Laptop/PC

Learning Goals	Learning Outcomes
1. Understand and apply JavaScript switch statements. 2. Use the 'break' keyword effectively within switch statements. 3. Implement the 'default' keyword in switch statements for unmatched conditions. 4. Apply multiple cases to the same code within switch statements. 5. Write and modify switch statements to manipulate variable values.	1. Understand and apply JavaScript switch statements to select code to run based on specific conditions. 2. Utilise the 'break' keyword to exit a switch statement after a condition has been met. 3. Implement the 'default' keyword to specify code to run when no case conditions are met. 4. Apply multiple case conditions to the same piece of code within a switch statement. 5. Write and modify a switch statement, including a default case, to manipulate variable values based on conditions.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Coding Short Course / Module: Advanced Coding

Advanced Coding

Lesson: Overview of how HTML, CSS, and JavaScript Interact

□ Beginner	□ 20 mins	□ Student Quiz

Prepare to explain the analogy of web development to building a house, with HTML, CSS, and JavaScript as the structure, design, and functionality respectively. Ensure understanding of the basic structures of HTML, CSS, and JavaScript, including their syntax and usage. Highlight the synergy of these three languages in creating dynamic web pages. Be ready to discuss real-life applications, such as creating a web-based quiz, to illustrate their interactivity. Conclude by emphasising the importance of proficiency in all three components for effective web development.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the roles of HTML, CSS, and JavaScript in web development. 2. Comprehend the basic structure and syntax of HTML, CSS, and JavaScript. 3. Apply CSS styles in different ways: inline, internal, and external. 4. Recognise common uses of JavaScript in enhancing web interactivity. 5. Appreciate the synergy of HTML, CSS, and JavaScript in creating dynamic web pages.	1. Understand the roles of HTML, CSS, and JavaScript in web development. 2. Identify the basic structure and elements of an HTML document. 3. Apply CSS to control the appearance of a webpage. 4. Use JavaScript to add interactivity to web pages. 5. Integrate HTML, CSS, and JavaScript to create dynamic web pages.

Lesson: Setting up Essential Tools

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

This lesson guides students through setting up essential web development tools. They'll learn about code editors, browser developer tools, and the console for debugging. Students will explore CodePen, an online code editor, and create a basic webpage using HTML, CSS, and JavaScript. They'll also add a button with an onclick event. The lesson concludes with a wrap-up and encouragement for further practice.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Gain proficiency in using web development environments and essential tools such as code editors, browser developer tools, and the console for debugging. 2. Understand the features and benefits of using CodePen as an online code editor. 3. Develop skills to inspect, debug, and optimise code using browser developer tools. 4. Learn to use the console for identifying and resolving issues in JavaScript code. 5. Apply knowledge to create a basic webpage on CodePen, incorporating HTML, CSS, and JavaScript, and adding interactive elements like buttons with onclick events.	1. Identify and utilise essential web development tools including code editors, browser developer tools, and the console for debugging. 2. Select and use CodePen as an online code editor for writing and previewing HTML, CSS, and JavaScript in real-time. 3. Inspect and debug code using browser developer tools and the console. 4. Create and edit a basic webpage on CodePen using HTML, CSS, and JavaScript. 5. Add interactive elements to a webpage, such as a button with an onclick event, using JavaScript.

Lesson: Scripting and DOM Manipulation

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through a hands-on exploration of scripting and DOM manipulation. They'll set up a CodePen project, create HTML structures, and add JavaScript functions. They'll learn to use 'onclick' attributes, event listeners, and manipulate text colour and size. The lesson concludes with challenges to create small text and remove elements, reinforcing their understanding of DOM manipulation.

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

- Chromebook/Laptop/PC

- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand and apply the concept of DOM manipulation using JavaScript. 2. Develop skills in creating and modifying HTML elements dynamically. 3. Gain proficiency in handling events using JavaScript, including click and mouseover events. 4. Learn to use JavaScript to alter CSS properties of HTML elements. 5. Apply problem-solving skills to complete coding challenges related to DOM manipulation.	1. Set up and utilise CodePen for HTML and JavaScript scripting. 2. Create and manipulate HTML structure using JavaScript. 3. Implement JavaScript functions to dynamically add elements to a webpage. 4. Utilise event listeners to trigger JavaScript functions. 5. Manipulate CSS properties of HTML elements through JavaScript.

Lesson: Dynamic Form Validation with JavaScript

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare for a hands-on lesson on dynamic form validation using JavaScript. Familiarise yourself with the CodePen environment, as students will be setting up their projects there. The lesson will guide students through creating a form, styling it with CSS, and adding JavaScript for validation. They will learn to validate fields for name, email, and password, ensuring the correct length and format. The lesson concludes with a challenge to add an age field and validate it.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand how to set up a project on CodePen for HTML, CSS, and JavaScript development. 2. Create and style a form using HTML and CSS. 3. Implement JavaScript code to validate form fields for specific requirements. 4. Test form validation and handle form submission using JavaScript. 5. Extend JavaScript validation to new form fields, demonstrating adaptability of skills.	1. Set up a project environment in CodePen. 2. Create a form with name, email, and password fields using HTML. 3. Style the form using CSS for better visual appeal. 4. Implement JavaScript code to prevent form submission and enable validation. 5. Validate the name field to ensure it is at least 3 characters long. 6. Validate the email field to ensure it contains '@' and '.' characters. 7. Validate the password field to ensure it is at least 8 characters long and contains at least one number and one letter. 8. Test the form by entering different values and checking if the validation works as expected. 9. Add an age field and validate it to ensure the person is over 13.

Lesson: Integrating External Libraries and APIs

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through the process of integrating external libraries and APIs into a project. The lesson involves setting up a project on CodePen, adding jQuery, creating an HTML structure, understanding jQuery syntax, selectors, and events, adding a click event listener, fetching weather data with an API, displaying the weather data, and extending the functionality of the weather app. The lesson concludes with wrapping up the weather app and encouraging students to explore more advanced features and APIs.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Develop proficiency in setting up a new project on CodePen. 2. Gain understanding and practical skills in integrating jQuery into a project. 3. Master the creation of HTML structures and the application of jQuery syntax and selectors. 4. Learn to handle jQuery events and implement event listeners. 5. Acquire skills in fetching data from external APIs and integrating it into a web application.	1. Set up a new project using CodePen and integrate jQuery library. 2. Create a basic HTML structure for a web application. 3. Understand and apply jQuery syntax to select and manipulate HTML elements. 4. Handle user interactions using jQuery event listeners. 5. Fetch and display real-time weather data from an external API.

Lesson: Interactive Quiz Game

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

This lesson guides students through creating an interactive quiz game using HTML, CSS, and JavaScript. They'll learn how to set up a project on CodePen, structure HTML for the game, style it with CSS, and add functionality with JavaScript. The lesson includes adding a jQuery library, setting up questions, variables, and functions to display questions, check answers, and display scores. It concludes with challenges to add a timer and different difficulty levels to the 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 setting up a coding project using CodePen. 2. Understand and apply HTML structure to create an interactive quiz game. 3. Apply CSS styling to enhance the visual presentation of the quiz game. 4. Utilise jQuery library to manipulate HTML elements and handle user interactions. 5. Create and manipulate JavaScript arrays and objects to store quiz questions and answers.	1. Create a new project on CodePen and add HTML structure for an interactive quiz game. 2. Apply CSS styling to HTML elements for visual enhancement. 3. Integrate the jQuery library into the project for dynamic features. 4. Set up an array of question objects and variables for tracking quiz progress. 5. Display questions and answer options dynamically, and check user's answers for correctness. 6. Implement functionality to move to the next question after an answer is selected. 7. Display the user's score after all questions have been answered. 8. Enhance the quiz game with a timer for each question. 9. Add different difficulty levels to the quiz game for varied user experience.

Lesson: Weather Web App

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

In this lesson, students will create a Weather Web App using HTML, CSS, and JavaScript. They will learn how to fetch and display real-time weather data using APIs and jQuery. The lesson will guide them through setting up the project, adding jQuery and Fontawesome, creating the HTML structure, styling the structure and headings, initializing JavaScript, fetching and displaying weather data, adding a unit toggle, and testing the app. They will also be encouraged to enhance their app by adding additional features such as a search bar and displaying more weather information.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Develop a functional weather web application using HTML, CSS, and JavaScript. 2. Utilise jQuery for efficient manipulation of HTML documents. 3. Integrate and use external libraries such as Fontawesome for enhanced visual appeal. 4. Fetch and display real-time weather data using APIs. 5. Implement a feature to toggle between Celsius and Fahrenheit temperature units.	1. Develop a Weather Web App using HTML, CSS, and JavaScript. 2. Integrate jQuery and Fontawesome libraries into a web project. 3. Construct HTML structure to display weather data. 4. Style the web app using CSS for an engaging user interface. 5. Fetch and display real-time weather data from an API using JavaScript.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Coding Short Course / Module: Advanced Coding

Advanced Coding

Lesson: An Introduction to Python

□ Intermediate	□ 30 mins	□ Student Quiz

Prepare to introduce Python as a beginner-friendly programming language, highlighting its use in various fields. Familiarise yourself with the Micro:bit Python editor for practical application. Discuss Python's syntax, particularly the importance of indentation and comments. Guide students through writing their first Python program and adding comments for clarity. Explain the sequence of code execution using a simple program. Encourage further practice and exploration post-lesson.

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 basics of Python programming language. 2. Utilise the Micro:bit Python editor for code writing and testing. 3. Comprehend and implement Python's indentation rules to define code blocks. 4. Use Python comments for code explanation and documentation. 5. Write, run, and debug simple Python programs using Micro:bit.	1. Understand and explain the basics of Python programming language and its application in various fields. 2. Access and navigate the Micro:bit Python editor for writing and testing Python code. 3. Apply Python indentation rules to define code blocks correctly. 4. Use Python comments to add notes and explanations to the code. 5. Write, run, and debug a simple Python program using the Micro:bit Python editor.

Lesson: Mastering Variables

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

This lesson guides students through the process of mastering variables in Python. They will learn about variable declaration, assignment, types, and naming conventions. The lesson also includes practical exercises such as creating a higher or lower game using the Microbits Python editor. Teachers should ensure students understand the concept of variables, their types, and how to manipulate them. They should also facilitate the game creation exercise, helping students apply their knowledge in a practical context.

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 concept of variables in Python, including their declaration, assignment, and types. 2. Learn and apply good variable naming conventions in Python. 3. Manipulate variable values through operations such as incrementing, decrementing, and string concatenation. 4. Apply knowledge of variables in creating a simple higher or lower game using the Microbits Python editor. 5. Gain familiarity with importing and using libraries in Python, specifically for game development.	1. Understand and apply the concept of variables in Python, including declaration, assignment, and types. 2. Manipulate variable values through incrementing, decrementing, and string concatenation. 3. Adhere to good naming conventions for variables, specifically snake_case and camelCase. 4. Import and utilise libraries in Python, specifically for the creation of a higher or lower game. 5. Develop a simple higher or lower game using variables, loops, conditionals, and libraries in Python.

Lesson: Looping Around

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

This lesson will guide students through understanding loops in Python, focusing on while loops, for loops, and nested loops. They will learn how Python uses indentation to define code blocks and how to break out of loops. The lesson includes practical exercises to reinforce learning, such as creating a project to measure reaction times. Teachers should ensure students understand the importance of consistent indentation and how loops can be used to control the flow of a program.

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
- Understand the importance of indentations in Python and how they define code blocks. - Learn how to use 'while' loops and 'for' loops to repeat code execution based on conditions. - Explore nested loops and how they can be used to iterate through multiple dimensions. - Learn how to break out of loops using the 'break' statement when a specific condition is met. - Apply the knowledge of loops and control structures to create a simple reaction time game. - Develop problem-solving skills by modifying and extending the provided code examples.	- By the end of this lesson, students will be able to identify and differentiate between while loops, for loops, and nested loops in Python. - Students will be able to demonstrate the use of proper indentation in Python code to define code blocks. - Students will be able to create and manipulate while loops and for loops to execute a block of code multiple times. - Students will be able to implement nested loops to control the flow of their program through multiple levels of iteration. - Students will be able to use the 'break' statement to exit a loop prematurely based on a specific condition. - Students will be able to apply their knowledge of loops and control structures to create a simple reaction time game using the micro:bit's LED matrix and buttons.

Lesson: Making Decisions

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through understanding conditional statements in MicroPython, including 'if', 'elif', and 'else'. They will create a simple project to reinforce their understanding, and then apply these concepts to a Dice Roller project. Ensure students understand how to use the Microbit Python Editor, and are comfortable with concepts such as loops, conditions, and using the micro:bit's accelerometer. Encourage experimentation with different conditions and scenarios to deepen their understanding.

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. Comprehend and apply conditional statements in coding, including 'if', 'elif', and 'else'. 2. Develop a basic project using 'if' statements to demonstrate understanding. 3. Understand and implement 'elif' and 'else' statements to create complex decision-making structures. 4. Grasp the concept of nested 'if' statements and their application in coding. 5. Create a Dice Roller project utilising 'if', 'elif', and 'else' statements, demonstrating the ability to make decisions in code based on specific conditions.	1. Apply conditional statements in Python code, specifically `if`, `elif`, and `else` statements. 2. Construct a simple `if` statement to check a condition and execute a block of code. 3. Create complex decision-making structures using `elif` and `else` statements in conjunction with `if` statements. 4. Utilise nested `if` statements to check multiple conditions within a single `if` statement. 5. Develop a Dice Roller project using the micro:bit's accelerometer, random number generation, and conditional statements to display different outcomes.

Lesson: Operators Decoded

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through understanding comparison and logical operators, and conditional Booleans in MicroPython. They'll apply these concepts in a practical project, creating a temperature indicator using the Microbit's online editor. Ensure they understand how to use these operators in 'if' and 'elif' statements. Encourage experimentation with different values to see how it affects conditions. The final project will involve using comparison and logical operators to determine temperature ranges and display appropriate images.

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 comparison operators in MicroPython. 2. Understand and apply logical operators in MicroPython. 3. Understand and utilise conditional Booleans in MicroPython. 4. Create a temperature indicator project using MicroPython and Micro:bit. 5. Apply knowledge of operators and conditional Booleans in practical coding scenarios.	1. Understand and apply comparison operators in MicroPython. 2. Utilise logical operators to combine conditional statements in MicroPython. 3. Implement conditional Booleans to make decisions based on the result of a condition. 4. Create a temperature indicator project using the built-in temperature sensor of Micro:bit. 5. Apply comparison and logical operators to determine temperature range and display appropriate images on Micro:bit.

Lesson: Array Essentials

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

In this lesson, teachers will guide students through the basics of working with arrays in MicroPython, using the micro:bit Python editor. Students will learn what an array is, how to create a list, retrieve and change list elements, add and remove elements from a list. The lesson culminates in a project where students will use arrays to create patterns of lights on the micro:bit LED display. 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

Required equipment for this lesson:

- Microbit

Learning Goals	Learning Outcomes
1. Understand and apply the concept of arrays in MicroPython. 2. Create, retrieve, and modify elements in a list. 3. Add and remove elements from a list. 4. Use arrays to create patterns of lights on the micro:bit LED display. 5. Combine and manipulate multiple arrays to create complex data structures.	1. Understand and define arrays in MicroPython. 2. Create, retrieve, and manipulate elements in a list. 3. Add and remove elements from a list using append(), extend(), remove(), and pop() methods. 4. Use arrays to store and manage data in MicroPython programs. 5. Apply array manipulation skills to create an LED light pattern project on a micro:bit.

Lesson: Advanced Array Tactics

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through advanced operations on Python lists using MicroPython. The lesson covers sorting lists in ascending and descending order, finding the length of a list, counting occurrences in lists, and applying these skills to create a strong password generator. Ensure students understand the use of sort(), len(), and count() methods, and how to use loops and random.choice() function. Encourage them to experiment with the code in their own projects.

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. Master advanced operations on lists in Python, including sorting in ascending and descending order. 2. Understand how to determine the length of a list using the len() function. 3. Learn to count the occurrences of a specific item in a list using the count() method. 4. Apply the learned concepts in a practical project to create a strong password generator. 5. Develop skills in manipulating and analysing data stored in lists.	1. Sort a list in ascending order using the sort() method in MicroPython. 2. Sort a list in descending order using the sort(reverse=True) method in MicroPython. 3. Determine the length of a list using the len() function in MicroPython. 4. Count occurrences of a specific item in a list using the count() method in MicroPython. 5. Generate a strong, random password using a combination of character sets in MicroPython.

Lesson: Function Junction

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

In this lesson, students will delve into Python programming, focusing on procedures and functions. They will learn the difference between the two, create simple procedures and functions using the micro:bit, and understand the use of parameters in functions. The lesson culminates in a project where students design a simplified weather station using their new skills. This hands-on approach will help reinforce their understanding of procedures and functions in Python.

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 difference between procedures and functions in Python programming. 2. Create and utilise procedures in Python code. 3. Create and utilise functions in Python code, including those that return values. 4. Develop functions with parameters to enhance flexibility and functionality. 5. Apply knowledge of procedures and functions to create a simple weather station project using MicroPython and Micro:bit.	1. Differentiate between procedures and functions in Python programming. 2. Create and utilise a procedure in Python to display a smiley face on the micro:bit's display. 3. Develop a function in Python that returns the square of a number and display the result on the micro:bit. 4. Construct a function with parameters in Python that takes two numbers and returns their sum, displaying the result on the micro:bit. 5. Design and implement a simplified weather station on the micro:bit using procedures and functions.

Lesson: Scope Showdown: Local vs. Global

□ Expert	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through understanding the concept of local and global variables in programming. Start with an introduction to the term 'scope', followed by a detailed explanation of local variables using Python code. Then, introduce global variables and their usage. Discuss best practices for using global variables. The lesson culminates in a practical project where students create a temperature logger using MicroPython, applying their understanding of local and global variables. Finally, wrap up the lesson by reinforcing the importance of variable scope in coding projects.

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 concept of variable scope in programming. 2. Distinguish between local and global variables and their usage. 3. Apply the concept of local and global variables in Python programming. 4. Adhere to best practices when using global variables. 5. Develop a Micro:bit temperature logger project using both local and global variables.	1. Understand and differentiate between local and global variables in programming. 2. Identify the scope of a variable and its accessibility within a program. 3. Apply the concept of local variables within a function, demonstrating their limited scope. 4. Utilise global variables appropriately within a program, demonstrating their wider scope. 5. Combine the use of local and global variables in a practical project, demonstrating understanding of best practices.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Coding Short Course / Module: Classroom Based Assessment (CBA)

Classroom Based Assessment (CBA)

Lesson: Project Planning

□ Intermediate	□ 120 mins

Emphasise the importance of planning in coding projects. Guide students through brainstorming program ideas, encouraging creativity. Assist in sketching designs and storyboarding to visualise the program's flow. Help list necessary resources, including assets and coding techniques. Facilitate creating a realistic timeline with task assignments. Conclude by reviewing and refining the plan, ensuring it's clear and adaptable.

Required equipment for this lesson:

- Pen & Paper

Learning Goals	Learning Outcomes
1. Develop effective project planning skills for coding projects. 2. Generate and select creative program ideas. 3. Design and storyboard program structures visually. 4. Identify and list necessary resources for program development. 5. Create and manage a project timeline with realistic deadlines.	1. Generate at least three program ideas and select one for development. 2. Create a storyboard that outlines the user interface and program flow. 3. List all required assets and coding techniques for the chosen program. 4. Develop a project timeline with tasks, time estimates, and deadlines. 5. Review and adjust the project plan to ensure clarity and achievability.

Lesson: Project Development

□ Intermediate	□ 320 mins

Begin with an introduction to the coding phase, encouraging students to use tools like Scratch or Microbits. Guide them through setting up their workspace, coding main features, testing, and adding enhancements. Emphasise saving work regularly. Conclude with final testing, saving the project, and preparing for presentation. Encourage problem-solving and note-taking throughout the process.

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

- Chromebook/Laptop/PC

Learning Goals	Learning Outcomes
1. Develop proficiency in setting up and managing a coding environment. 2. Acquire skills in coding core functionalities of a program. 3. Master the process of testing and debugging code. 4. Enhance programs with additional features and document progress. 5. Finalise and prepare a program for presentation.	1. Set up a coding environment using Scratch or Microbit, saving and reopening projects. 2. Code core program functionality, adding features incrementally and testing each addition. 3. Identify and fix errors in the program through systematic testing and debugging. 4. Enhance the program with additional features like sound effects, score counters, or animations. 5. Finalise the program, conduct a final test, and prepare documentation for presentation.

Lesson: Project Presentation and Evaluation

Intermediate	240 mins

Prepare students for presenting their programs, focusing on clarity and brevity. Ensure they practice demonstrating features and prepare notes on program functionality, coding techniques used, and challenges overcome. Facilitate a supportive environment for feedback and reflection, celebrating all students' achievements and encouraging them to share their work.

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

- Chromebook/Laptop/PC

Learning Goals	Learning Outcomes
1. Develop effective presentation skills for showcasing software projects. 2. Enhance communication of technical concepts and coding techniques used in projects. 3. Learn to receive and utilise constructive feedback for personal and project improvement. 4. Reflect on project experiences to identify learning outcomes and areas for growth. 5. Celebrate achievements and foster a sense of accomplishment in programming endeavours.	1. Deliver a 3-5 minute presentation on their program, covering its function, coding techniques used, and a challenge overcome. 2. Demonstrate the program's key features effectively to an audience. 3. Respond to audience questions using prepared notes and knowledge of the project. 4. Collect and record feedback from the audience on what was liked and areas for improvement. 5. Reflect on the project, identifying successes, challenges faced, and lessons learned.

Digital Media Literacy Short Course

This short course introduces students to essential digital media concepts and skills across various engaging modules. As a teacher, start with a welcoming tone to ease students into discussions. Utilise provided visuals and interactive activities to connect lessons to their daily online experiences. Encourage reflection and critical thinking to deepen understanding and ensure practical application of concepts.

Duration	Equipment
Classroom hours ~71.33333333333333	Students can use any of these devices: • Chromebook/Laptop/PC • iPad/Tablet Required Equipment: • Headphones • Webcam/camera
Learning Goals	**Learning Outcomes**
1. Understand the fundamental concepts and importance of digital media in everyday life. 2. Develop skills to navigate online spaces safely and responsibly. 3. Acquire proficiency in using digital tools for communication, creativity, and learning. 4. Evaluate the credibility and reliability of online information and sources. 5. Foster critical thinking and ethical considerations in digital content creation and interaction.	1. Define digital media and explain its significance in personal and societal contexts through reflective activities. 2. Identify and select appropriate digital tools for specific tasks in communication, creativity, and learning. 3. Demonstrate safe and responsible navigation of online spaces by distinguishing between secure and risky behaviours. 4. Evaluate the credibility of online sources using structured methods like the CRAAP Test to ensure reliable information use. 5. Create a multimedia project by integrating images, audio, and video, applying ethical editing practices with digital tools.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Digital Media Literacy Short Course / Module: Digital Media Basics

Digital Media Basics

Lesson: What is Digital Media?

□ Beginner	□ 40 mins	□ Student Quiz

These teaching notes are designed to support educators in delivering an introductory session on digital media. Begin by welcoming students and setting a relaxed tone. Use the provided visuals to spark discussion on everyday digital content. Encourage reflection through activities, guiding students to connect concepts to their personal online experiences.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the definition and scope of digital media in everyday life. 2. Recognise the importance of digital media literacy for safe and effective online engagement. 3. Identify examples of digital media and their relevance to personal experiences. 4. Explore the impact of digital media on communication, learning, and creativity. 5. Develop an awareness of responsible and ethical online behaviour.	1. Define digital media and identify at least three examples of it in everyday life. 2. Explain the importance of digital media literacy for staying safe and responsible online. 3. List three specific skills or topics covered in the course related to digital media use. 4. Provide three personal examples of digital media use and describe why literacy is important for each. 5. Reflect on personal digital media habits by answering guided questions about online behaviour and responsibility.

Lesson: Digital Tools Overview

□ Beginner	□ 40 mins	□ Student Quiz	□ Student Challenge

These teaching notes are designed to support educators in delivering an engaging lesson on digital tools. Begin with a warm welcome, encouraging students to think about their daily tech use. Guide them through definitions, categories, and practical examples using the provided visuals. Facilitate discussions prompted by reflection questions and oversee activities to ensure students apply concepts by matching tools to tasks.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the concept and purpose of digital tools in everyday tasks. 2. Identify different categories of digital tools and their specific uses. 3. Evaluate and select appropriate digital tools for various scenarios. 4. Recognise the importance of safety and efficiency when using digital tools. 5. Apply digital tools effectively within the context of course activities and projects.	1. Define digital tools and provide at least two examples of their use in daily activities. 2. Identify and describe four main categories of digital tools, including one specific tool for each category. 3. Select an appropriate digital tool for a given scenario and justify the choice based on task requirements. 4. Demonstrate understanding of how specific digital tools will be used within the course for tasks like writing, creating, or collaborating. 5. Complete a matching activity by pairing tasks with suitable digital tools and explaining the reasoning behind each selection.

Lesson: Navigating Online Spaces

□ Beginner	□ 40 mins	□ Student Quiz	□ Student Challenge

These teaching notes are designed to support educators in delivering a lesson on safe internet use. Begin with an engaging introduction to online environments, then guide students through identifying different types of digital spaces. Use interactive scenarios to discuss safe versus risky behaviours, encouraging reflection. Facilitate activities like the scavenger hunt to reinforce practical skills.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the concept and purpose of various online spaces. 2. Identify different types of online environments and their unique characteristics. 3. Distinguish between safe and risky behaviours in digital interactions. 4. Apply responsible decision-making skills in online scenarios. 5. Develop strategies to navigate online spaces securely and confidently.	1. Define online spaces and identify at least three distinct types with specific examples. 2. Distinguish between safe and risky behaviours in online environments, listing two examples of each. 3. Apply safe online practices by responding appropriately to a given scenario involving personal information requests. 4. Evaluate the trustworthiness of a website by identifying one safety indicator during a guided activity. 5. Demonstrate decision-making skills by selecting safe responses in multiple online interaction scenarios.

Lesson: Digital Communication Basics

□ Beginner	□ 40 mins	□ Student Quiz	□ Student Challenge

To prepare for this lesson, familiarise yourself with the core concepts of online interaction. Focus on defining digital communication, exploring various tools and styles, and emphasising respectful, clear, and safe behaviours. Encourage student engagement through reflective questions and practical activities like rewriting messages to reinforce effective communication skills.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the fundamental concepts and importance of digital communication in daily life. 2. Identify various tools and styles of digital communication for different contexts. 3. Apply respectful and clear communication behaviours in online interactions. 4. Recognise safety practices to protect privacy and maintain security online. 5. Reflect on personal communication styles to improve clarity and effectiveness.	1. Define digital communication and explain its importance in daily life. 2. Identify at least three different tools for digital communication and describe their appropriate contexts for use. 3. Demonstrate the difference between casual and formal communication styles in written examples. 4. Apply principles of tone, clarity, and digital etiquette to rewrite a given message for improved effectiveness. 5. Reflect on personal online communication habits and suggest one area for improvement in clarity or respect.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Digital Media Literacy Short Course / Module: Digital Media Basics

Digital Media Basics

Lesson: The World of the Internet

□ Beginner	□ 30 mins	□ Student Quiz	□ Student Challenge

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 the Internet, its history, and its significance in the modern world. • Comprehend how the Internet works, including the process of sending and receiving information. • Explore the various uses of the Internet, including messaging, information sharing, and accessing websites. • Learn about the structure and purpose of URLs and how they are used to access websites. • Recognize the importance of responsible and respectful behavior when using the Internet. • Appreciate the role of the Internet in connecting people globally and facilitating information exchange.	• By the end of this lesson, students will be able to define what the Internet is and explain its importance. • Students will be able to describe how the Internet works, including how information is sent and received. • Students will be able to identify different ways to use the Internet, such as sending messages and sharing information. • Students will be able to explain how messages are sent over the Internet, using the concept of 'data packets'. • Students will be able to identify different platforms for sharing information on the Internet, including websites, blogs, and social media. • Students will be able to understand and explain what a URL is and how it is used to access websites.

Lesson: How Does the Internet Work?

□ Beginner	□ 30 mins	□ Student Quiz	□ Student Challenge

Prepare by familiarising yourself with internet components like devices, servers, ISPs, and routers. Use a relatable scenario of sending a message to explain the journey from sender to receiver. Engage students by letting them choose their message type. Highlight the roles of each internet component in the message's travel. Conclude with a summary of the process, emphasising packet transmission and reassembly.

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 components and their roles in the Internet's operation. 2. Recognise how data travels from sender to receiver across the Internet. 3. Appreciate the process of message transmission, including packetisation and routing. 4. Develop a conceptual model of Internet communication using a narrative approach. 5. Gain insight into the collaborative nature of Internet technology.	1. Identify and describe the roles of key Internet components (Sender, Receiver, Post Offices, Delivery Team, Traffic Managers) in message transmission. 2. Explain the process of sending a message from a Sender to a Receiver via the Internet. 3. Describe how messages are broken into packets and reassembled at the destination. 4. Illustrate the journey of a message through servers and routers using a narrative. 5. Summarise the overall function of the Internet in connecting devices for communication.

Lesson: Different Types of Devices

□ Beginner	□ 30 mins	□ Student Quiz	□ Student Challenge

Lesson Title: Different Types of Devices Teaching Notes: 1. Prepare a presentation covering the different types of devices: computers, smartphones, gaming consoles, smart TVs, smartwatches, e-readers, and smart home devices. Include images and key features of each device. 2. Be ready to explain how each device connects to the internet and its uses. 3. Prepare questions to engage students during the lesson, such as asking them to name devices they use at home or school. 4. For the concluding activity, ensure students understand the task. If possible, provide a worksheet for them to record their findings. 5. Encourage students to share their findings and discuss the importance of these devices in their daily lives.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• Understand the basic functions and characteristics of various digital devices such as computers, smartphones, gaming consoles, smart TVs, smartwatches, e-readers, and smart home devices. • Recognize the significance of internet connectivity in these devices and how it enhances their functionality. • Identify the differences and similarities between these devices in terms of their use, portability, and connectivity. • Appreciate the role of these devices in our daily lives, from communication and entertainment to information access and home automation. • Develop the ability to critically analyze the suitability of different devices for different tasks or needs. • Apply the knowledge gained by identifying and counting the different devices in their own environment.	• Identify and describe the functions of different types of devices such as computers, smartphones, gaming consoles, smart TVs, smartwatches, e-readers, and smart home devices. • Explain how each type of device connects to the internet and why this connection is important. • Differentiate between the various types of devices based on their uses, features, and connectivity options. • Recognize popular brands and models within each type of device category. • Apply knowledge of devices to identify and count the number of different devices in a given environment. • Communicate findings and share knowledge about different types of devices with peers or family members.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Digital Media Literacy Short Course / Module: Digital Media Basics

Digital Media Basics

Lesson: What is Personal Information?

□ Beginner	□ 30 mins	□ Student Quiz	□ Student Challenge

This lesson aims to educate students about the concept of personal information, its importance, and the potential risks associated with sharing it online. Teachers should prepare examples of personal information and discuss why it's crucial to keep such information private. The lesson should also cover the dangers of sharing personal information on the internet and provide practical tips for protecting personal information. Teachers should encourage students to reflect on their online habits and consider changes to enhance their online safety. Reinforce the importance of privacy and caution when interacting online.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• Understand the definition of personal information and why it is important. • Identify examples of personal information. • Understand the risks associated with sharing personal information online. • Learn strategies for protecting personal information on the internet. • Reflect on personal online habits and identify areas for improvement in protecting personal information. • Develop an attitude of caution and responsibility when it comes to sharing personal information online.	• Students will be able to define what personal information is and provide examples of it. • Students will understand the importance of personal information and the potential risks associated with sharing it. • Students will be able to explain the relationship between personal information and internet safety. • Students will learn and be able to list strategies for protecting their personal information online. • Students will be able to assess their own online habits and identify areas where they can improve their personal information security. • Students will understand the importance of keeping their personal information private as a part of staying safe online.

Lesson: Why We Shouldn't Share Personal Information Online

□ Beginner	□ 30 mins	□ Student Quiz	□ Student Challenge

This lesson aims to educate students on the importance of online privacy and the risks associated with sharing personal information on the internet. Teachers should prepare real-life examples of each risk (identity theft, cyberbullying, unwanted contact, phishing scams) to help students understand the implications. Encourage students to share their experiences and thoughts on each topic. For younger students, consider role-playing scenarios to demonstrate how to handle such situations. The lesson should conclude with a discussion on the importance of reporting any online harassment or suspicious activity to a trusted adult.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• Understand what constitutes personal information and the potential risks associated with sharing it online. • Identify and comprehend the various risks of sharing personal information online, including identity theft, cyberbullying, unwanted contact from strangers, and phishing scams. • Recognize the implications of identity theft and how sharing personal information online can lead to such a situation. • Understand the concept of cyberbullying and how sharing personal information online can make one more vulnerable to it. • Learn how to avoid unwanted contact from strangers by being cautious about sharing personal information online. • Understand what phishing scams are and how they can be avoided by not sharing personal information online.	• Students will be able to define what constitutes personal information and understand the risks associated with sharing it online. • Students will be able to explain the concept of identity theft and how it can be facilitated by sharing personal information online. • Students will be able to describe the phenomenon of cyberbullying and understand how sharing personal information can make one vulnerable to it. • Students will be able to identify the dangers of unwanted contact from strangers online and how it can be avoided by not sharing personal information. • Students will be able to recognize phishing scams and understand how they can be used to trick individuals into sharing personal information. • Students will be able to apply the knowledge gained to protect their personal information and maintain online safety.

Lesson: Strong vs. Weak Passwords

□ Beginner	□ 30 mins	□ Student Quiz	□ Student Challenge

In this lesson, teachers will introduce the concept of passwords and their importance in online security. They will explain the difference between weak and strong passwords, using examples to illustrate. Teachers will then delve into the role of password length and character variety in enhancing password strength, using a table to demonstrate the time it would take to guess different types of passwords. The lesson concludes with a recap and an activity where students create their own strong passwords. Teachers should ensure students understand the importance of password security and the elements that constitute a strong password.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
- Understand the importance of using strong passwords for online security. - Identify the characteristics of weak passwords and why they are easy to guess. - Understand the components of a strong password and how to create one. - Comprehend how the length and complexity of a password affects its security. - Analyze the strength of different types of passwords based on their composition and length. - Apply the learned knowledge to create and use strong, secure passwords in real-world contexts.	- Understand the importance of strong passwords and the risks associated with weak passwords. - Identify examples of weak passwords and explain why they are easy to guess or figure out. - Identify the characteristics of strong passwords and explain how they increase account security. - Understand how the length and complexity of a password affect its strength and the likelihood of it being hacked. - Analyze and compare the strength of different types of passwords based on their length and the types of characters used. - Create a strong password using the guidelines provided in the lesson.

Lesson: The Dangers of Sharing Passwords

□ Beginner	□ 30 mins	□ Student Quiz	□ Student Challenge

In this lesson, students will learn about the importance of password security. Teachers should emphasize the comparison of passwords to house keys, reinforcing the idea of privacy and security. The lesson will discuss the potential risks of password sharing, including identity theft and account misuse. Teachers should encourage students to never share passwords, even with friends, and to report any pressure to do so. The lesson concludes with practical tips for password protection. Teachers may want to role-play scenarios where students have to refuse password requests. Reinforce the importance of seeking adult guidance in uncertain situations.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
- Understand the purpose and importance of passwords in protecting online accounts and personal information. - Recognize the potential dangers and consequences of sharing passwords. - Identify situations and individuals with whom passwords should never be shared. - Know how to respond if someone asks for their password. - Learn strategies for protecting and maintaining the security of their passwords. - Develop an attitude of responsibility and caution when it comes to handling their personal online security.	- Students will be able to explain the importance and purpose of passwords in protecting personal information and online accounts. - Students will be able to identify the potential dangers and consequences of sharing passwords. - Students will be able to list the individuals with whom they should never share their passwords. - Students will be able to describe appropriate actions to take if someone asks for their password. - Students will be able to demonstrate understanding of how to protect their passwords effectively. - Students will be able to understand the importance of seeking guidance from a trusted adult when faced with a situation involving password security.

Lesson: What is a Digital Footprint?

□ Intermediate	□ 30 mins	□ Student Quiz	□ Student Challenge

These teaching notes are for a lesson on 'What is a Digital Footprint?'. Teachers should prepare by familiarizing themselves with the concept of digital footprints and the difference between active and passive footprints. They should also consider examples of how digital footprints can impact online reputation, privacy, safety, future opportunities, and relationships. Teachers should encourage students to critically evaluate their own online behavior and consider the permanence of their digital footprints. The lesson should conclude with a reflection on how students can improve their digital footprints and protect their online reputation and privacy.

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 a digital footprint and its implications on online activity. • Identify the two types of digital footprints: active and passive, and understand how they are created. • Recognize the permanence of digital footprints and the difficulty in completely erasing online information. • Understand the impact of digital footprints on online safety, future opportunities, and personal relationships. • Develop a critical thinking approach towards online posting to protect online reputation and privacy. • Reflect on personal online activities and consider ways to improve and protect their digital footprint.	• Students will be able to define what a digital footprint is and understand its permanence. • Students will be able to differentiate between active and passive digital footprints and provide examples of each. • Students will understand the impact of their digital footprint on their online reputation, privacy, safety, future opportunities, and relationships. • Students will be able to apply strategies to manage their digital footprint, including the principle of 'thinking before posting'. • Students will be able to reflect on their current digital footprint and identify areas for improvement. • Students will understand the importance of maintaining a positive digital footprint and how it can affect their personal and professional life.

Lesson: How a Digital Footprint is Created

□ Intermediate	□ 30 mins	□ Student Quiz	□ Student Challenge

These teaching notes will guide teachers through a lesson on digital footprints. The lesson begins with an introduction to the concept of a digital footprint, followed by a discussion on the actions that contribute to it. Teachers should provide examples of digital footprints and differentiate between positive and negative ones. The lesson then moves on to strategies for creating a positive digital footprint and avoiding a negative one. Teachers should emphasize that a digital footprint is always growing and encourage students to reflect on their online behavior. The lesson concludes with a review and reflection activity.

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 a digital footprint and how it is created through online activities. • Identify examples of actions that contribute to a digital footprint. • Differentiate between positive and negative digital footprints and their potential impacts. • Develop strategies for creating a positive digital footprint and avoiding a negative one. • Recognize that a digital footprint is continuously growing and can have long-term effects. • Reflect on personal online behaviors and consider ways to improve and maintain a positive digital footprint.	• Understand and explain what a digital footprint is and how it is created through online activities. • Identify and give examples of actions that contribute to a digital footprint. • Differentiate between positive and negative digital footprints and provide examples of each. • Demonstrate knowledge of how to create a positive digital footprint and avoid a negative one. • Understand that a digital footprint is continuously growing and reflect on the importance of being aware of one's online actions. • Reflect on their own digital footprint and identify ways to improve it.

Lesson: Think Before You Post

□ Intermediate	□ 30 mins	□ Student Quiz	□ Student Challenge

This lesson is about the importance of thinking before posting online. Teachers should emphasize the potential long-term consequences of sharing content online, such as damage to reputation or legal issues. Use real-life examples to illustrate these points. The T.H.I.N.K. method (True, Helpful, Inspiring, Necessary, Kind) should be introduced as a tool for students to use before they post. Encourage students to consider the feelings of others and the wide audience that can view their posts. The lesson should conclude with a reminder of the importance of maintaining a positive digital footprint.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• Understand the importance of thinking before posting anything online and the impact it can have on one's digital footprint and online reputation. • Recognize the potential negative consequences of not thinking before posting, including embarrassment, damage to online reputation, and legal trouble. • Learn and apply the T.H.I.N.K. method (True, Helpful, Inspiring, Necessary, Kind) before sharing content online. • Develop empathy and consideration for the feelings of others when posting content online. • Understand that what is posted online can be seen by a wide audience, including friends, family, teachers, and strangers. • Develop a habit of pausing and reflecting before hitting 'post' or 'send' to ensure comfort with the content being shared.	• Students will be able to explain the importance of thinking before posting online and the potential consequences of not doing so. • Students will be able to identify examples of negative consequences that can occur from not thinking before posting online. • Students will be able to apply the T.H.I.N.K. method to evaluate potential posts for appropriateness and potential impact. • Students will be able to demonstrate an understanding of how to maintain a positive online presence and protect their digital footprint. • Students will be able to evaluate a given scenario and determine whether or not it is appropriate to post online based on the T.H.I.N.K. method. • Students will be able to demonstrate the ability to pause and consider the potential impact before posting content online.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Digital Media Literacy Short Course / Module: Digital Media Basics

Digital Media Basics

Lesson: Crafting Effective Search Queries

□ Beginner	□ 40 mins	□ Student Quiz	□ Student Challenge

These teaching notes are designed to support educators in guiding students through a step-by-step exploration of online search techniques. Encourage interactive discussions on the importance of precise queries, demonstrate keyword usage, and facilitate hands-on activities like rewriting poor searches. Use provided examples to illustrate concepts and ensure students experiment with search engines for practical learning.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the importance of effective search skills in accessing relevant online information quickly. 2. Define and construct a search query using clarity, specificity, and relevance. 3. Differentiate between keywords and full questions to optimise search results. 4. Apply practical strategies and advanced techniques to refine and enhance search queries. 5. Practise rewriting and improving search queries to achieve more accurate and focused outcomes.	1. Construct clear and specific search queries using keywords to retrieve relevant online information. 2. Differentiate between full questions and keyword-based queries to improve search result accuracy. 3. Apply practical strategies, such as prioritising key terms and avoiding filler words, to refine search effectiveness. 4. Utilise advanced search techniques, including quotation marks, minus signs, and site-specific operators, to filter results precisely. 5. Rewrite ineffective search queries to enhance clarity and relevance through hands-on practice.

Lesson: Identifying Reliable Sources

□ Beginner	□ 40 mins	□ Student Quiz	□ Student Challenge

Prepare by reviewing the characteristics of reliable and unreliable sources, focusing on accuracy, authorship, and red flags like clickbait or bias. Facilitate discussions using the provided checklist to evaluate sample headlines. Encourage students to reflect on their trusted sources and apply critical thinking during practical activities for effective learning.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the importance of source reliability in online research. 2. Identify key characteristics of reliable and unreliable sources. 3. Recognise common red flags indicating poor-quality information. 4. Apply a structured checklist to evaluate the credibility of sources. 5. Develop critical thinking skills through practical evaluation and reflection.	1. Identify the key characteristics of reliable and unreliable sources. 2. Recognise common red flags indicating poor-quality or biased information. 3. Apply a five-question checklist to evaluate the credibility of online content. 4. Analyse sample headlines to determine their reliability using specific criteria. 5. Reflect on personal research habits and propose improvements for future searches.

Lesson: Search and Evaluate

□ Beginner	□ 40 mins	□ Student Quiz	□ Student Challenge

These teaching notes are designed to support educators in guiding students through a structured process of online research. Begin with an engaging introduction to the importance of systematic searching. Use the provided workflow and checklist to demonstrate evaluation techniques. Facilitate hands-on activities and encourage reflection to reinforce learning and build confidence.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the structured process of conducting research from formulating a question to selecting a reliable source. 2. Develop skills to craft effective search queries for accurate and relevant results. 3. Apply evaluation criteria to assess the credibility and reliability of online sources. 4. Compare multiple sources to identify the most trustworthy and relevant information. 5. Reflect on personal research practices to improve future digital information skills.	1. Formulate a precise search query from a research question to retrieve relevant information. 2. Apply a structured checklist to evaluate the credibility of online sources based on authorship, accuracy, and purpose. 3. Compare multiple sources using criteria such as recency, depth, and bias to select the most reliable one. 4. Conduct an independent search and document the evaluation process to justify the chosen source. 5. Reflect on the search and evaluation workflow to identify strengths and areas for improvement in future research tasks.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Digital Media Literacy Short Course / Module: Digital Media Basics

Digital Media Basics

Lesson: What is AI?

□ Beginner	□ 40 mins	□ Student Quiz

These teaching notes are designed to support educators in delivering an engaging lesson on artificial intelligence. Begin by encouraging students to reflect on familiar smart technologies to spark curiosity. Guide them through definitions, everyday examples, and basic AI processes using interactive activities like image quizzes. Facilitate discussions on AI’s capabilities and limits to foster critical thinking.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the fundamental definition and concept of Artificial Intelligence as a branch of computer science. 2. Identify examples of AI in everyday life and its practical applications across various domains. 3. Explain the basic process of how AI systems function, including data collection, pattern recognition, and prediction. 4. Recognise the capabilities and limitations of AI, distinguishing between human and machine intelligence. 5. Develop critical thinking skills to evaluate AI-generated content and its implications in digital contexts.	1. Define Artificial Intelligence (AI) and explain its core purpose in simulating human-like tasks. 2. Identify at least three examples of AI in everyday life, such as voice assistants or recommendation systems. 3. Describe the basic process of how AI works, including data collection, pattern recognition, and prediction. 4. Differentiate between AI-generated and real images through a guided activity, achieving at least 60% accuracy. 5. Outline one strength and one limitation of AI, demonstrating an understanding of its capabilities and boundaries.

Lesson: Basic Interactions

□ Beginner	□ 40 mins	□ Student Quiz	□ Student Challenge

These teaching notes are designed to support educators in delivering a lesson on fundamental AI interactions. Begin by engaging students with real-life examples of AI, like voice assistants. Guide them through each step, encouraging reflection and discussion. Use the provided activities to reinforce concepts, ensuring students connect theory to personal experiences for deeper understanding.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the fundamental concepts of interacting with artificial intelligence in everyday situations. 2. Identify common examples of AI tools and their applications in daily life. 3. Explain the basic mechanisms behind AI interactions, including inputs, outputs, and language processing. 4. Develop skills to communicate effectively with AI systems for accurate and helpful responses. 5. Recognise the importance of using AI safely and responsibly in various contexts.	1. Identify at least three everyday examples of AI interactions, such as voice assistants or autocomplete features, and explain their purpose. 2. Describe the basic process of how AI interprets inputs and generates outputs using concepts like training data and pattern recognition. 3. Differentiate between AI and human interactions in given scenarios by analysing response characteristics like speed and personalisation. 4. Demonstrate a simulated conversation with an AI system by creating a short script with clear inputs and expected outputs. 5. Apply safety and respect guidelines when interacting with AI by listing at least two specific practices, such as avoiding personal information sharing.

Lesson: Prompt Engineering

□ Beginner	□ 40 mins	□ Student Quiz	□ Student Challenge

These teaching notes are designed to support educators in delivering an engaging lesson on crafting effective interactions with AI tools. Begin by introducing the concept and its importance, then guide students through comparing weak and strong examples. Facilitate hands-on activities to improve prompts, encourage creativity, and discuss potential pitfalls. Conclude with reflective questions to reinforce learning.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the fundamental concept and importance of crafting effective prompts for AI interactions. 2. Distinguish between weak and strong prompts to improve the quality of AI responses. 3. Identify and apply key components of strong prompts, including task, topic, format, and tone. 4. Practise creating and refining prompts through hands-on activities and creative challenges. 5. Recognise potential limitations and risks of AI responses, applying critical thinking to mitigate issues.	1. Define prompt engineering and explain its importance in achieving accurate AI responses. 2. Distinguish between weak and strong prompts by identifying key differences in clarity and detail. 3. Construct strong prompts by incorporating the four key components: task, topic, format, and tone. 4. Improve weak prompts through practical exercises, ensuring specificity and relevance in rewritten versions. 5. Recognise common issues with AI responses, such as biases or inaccuracies, and apply strategies to mitigate them.

Lesson: Using AI for Web Searches

□ Beginner	□ 40 mins	□ Student Quiz	□ Student Challenge

Prepare students for a lesson on leveraging AI in online searches by encouraging reflection on their prior search experiences. Facilitate discussions on traditional versus AI-powered tools, guiding them through comparisons and hands-on activities. Use provided scenarios to spark critical thinking about tool selection, ensuring internet access for interactive tasks.

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

- Chromebook/Laptop/PC

- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the fundamental differences between traditional search engines and AI-powered search tools. 2. Identify the strengths and limitations of both search methods for various information needs. 3. Apply critical thinking to evaluate the accuracy and reliability of information from AI tools and search engines. 4. Determine the most appropriate search tool based on specific scenarios or purposes. 5. Develop strategies for verifying information using multiple sources to ensure trustworthiness.	1. Identify the differences between traditional search engines and AI-powered search tools in terms of functionality and output. 2. Explain the advantages and disadvantages of using AI tools versus traditional search engines for specific search purposes. 3. Apply critical thinking to select the appropriate tool (AI or traditional search engine) for various search scenarios based on reliability and need. 4. Conduct a comparative analysis of search results from both AI tools and traditional search engines to evaluate accuracy and presentation. 5. Demonstrate the ability to verify information from AI-generated responses by cross-referencing with credible sources.

Lesson: AI Q&A

□ Beginner	□ 40 mins	□ Student Quiz	□ Student Challenge

Prepare students for an engaging exploration of AI-powered question and answer tools by guiding them through each step. Encourage curiosity while emphasising responsible use. Facilitate discussions on AI's capabilities and limitations, and support hands-on activities to interact with tools like ChatGPT. Promote critical thinking and online safety throughout.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the fundamental concepts and functionalities of AI-powered question and answer tools. 2. Recognise appropriate contexts and limitations for using AI Q&A tools in learning and exploration. 3. Develop skills to interact effectively with AI tools through practical engagement and evaluation. 4. Identify common pitfalls and apply safe, responsible usage strategies when using AI resources. 5. Foster critical thinking by reflecting on AI interactions and verifying information with reliable sources.	1. Explain the fundamental processes behind AI Q&A tools, including parsing, data referencing, and natural language generation. 2. Identify appropriate scenarios for using AI Q&A tools, such as obtaining quick facts or clarifying concepts, while recognising limitations for sensitive topics. 3. Demonstrate the ability to interact with an AI Q&A tool by formulating and evaluating responses to factual, creative, and how-to questions. 4. Recognise common pitfalls of AI Q&A tools, such as hallucinations and outdated information, and apply strategies to verify accuracy. 5. Reflect on personal experiences with AI Q&A tools, articulating their potential uses and limitations in everyday learning contexts.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Digital Media Literacy Short Course / Module: Intermediate Digital Media

Intermediate Digital Media

Lesson: Digital Storytelling

□ Intermediate	□ 80 mins	□ Student Quiz	□ Student Challenge

These teaching notes are designed to support educators in delivering a 90-minute lesson on creating narratives with digital tools. Begin with an engaging discussion on memorable stories, guide students through comparisons of traditional and digital formats, and facilitate hands-on activities like story planning and pitching. Encourage reflection on ethical considerations and the impact of multimedia elements.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the concept and unique characteristics of digital storytelling compared to traditional methods. 2. Identify and apply key elements of effective digital narratives, including structure, visuals, and sound. 3. Analyse and evaluate examples of digital storytelling across various media formats. 4. Develop skills to plan and conceptualise an original digital story using multimedia tools. 5. Recognise the ethical considerations and responsibilities in creating digital content.	1. Define digital storytelling and distinguish it from traditional storytelling by identifying at least three unique multimedia elements. 2. Identify and describe five key components of an engaging digital story, including narrative structure, visuals, and sound. 3. Analyse a provided digital storytelling example, evaluating its effectiveness based on specific criteria like pacing and emotional impact. 4. Create a detailed plan for a digital story, incorporating characters, setting, plot, and digital tools in a structured template. 5. Demonstrate ethical considerations in digital storytelling by listing at least three ways to ensure respect and accuracy in content creation.

Lesson: Image Editing Basics

□ Intermediate	□ 80 mins	□ Student Quiz	□ Student Challenge

These teaching notes are designed to support educators in delivering a lesson on fundamental image editing skills. Begin by familiarising yourself with Canva, ensuring students have access to the platform. Guide them through each step, from basic concepts to practical tasks, encouraging creativity. Emphasise ethical editing practices and facilitate reflection activities to reinforce learning and responsibility.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Grasp the fundamental concepts and practical applications of image editing. 2. Master key techniques such as cropping, resizing, and applying filters using digital tools. 3. Develop skills to create and design original graphics with creativity and precision. 4. Understand and apply ethical principles in image editing to ensure authenticity. 5. Reflect on acquired skills to identify strengths and areas for improvement.	1. Apply basic image editing techniques such as cropping, resizing, and adjusting brightness and contrast using Canva. 2. Demonstrate the use of filters and layering to enhance the visual appeal of an image. 3. Export edited images in appropriate file formats like PNG or JPG for specific purposes. 4. Create a unique graphic by combining multiple editing skills to convey a clear theme or message. 5. Evaluate the ethical implications of image editing and apply responsible practices in digital content creation.

Lesson: Video Editing Intro

□ Intermediate	□ 80 mins	□ Student Quiz	□ Student Challenge

These teaching notes are designed to support educators in guiding students through an introductory video editing lesson using Clipchamp. Begin with a brief discussion on the purpose of video editing, then guide students step-by-step through key concepts like trimming, transitions, and adding effects. Encourage safe online practices and copyright awareness throughout the hands-on activities.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Grasp the fundamental concepts and importance of video editing for storytelling and content creation. 2. Master basic video editing techniques, including trimming, cutting, and using transitions. 3. Develop proficiency in using Clipchamp to edit and enhance videos with effects, text, and music. 4. Create a cohesive short video project by applying learned skills responsibly. 5. Understand the importance of copyright and safe online practices in digital content creation.	1. Define video editing and explain its importance in creating engaging digital content. 2. Identify and describe key video editing terms such as timeline, trim, cut, and transitions. 3. Demonstrate the use of Clipchamp to import, arrange, and edit video clips on a timeline. 4. Apply basic editing techniques including trimming, cutting, adding transitions, text, effects, and royalty-free music to a short video project. 5. Export a completed video project in an appropriate format and resolution using Clipchamp.

Lesson: Audio in Digital Media

□ Intermediate	□ 80 mins	□ Student Quiz	□ Student Challenge

These teaching notes are designed to support educators in delivering a lesson on the significance of sound in digital content creation. Begin by familiarising yourself with AudioMass, the free online editing tool used in the lesson. Encourage student engagement through hands-on activities like trimming and layering audio. Guide discussions on audio's role in storytelling, ensuring students source royalty-free resources responsibly. Headphones for this lesson are recommended.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Required equipment for this lesson:

- Headphones

Learning Goals	Learning Outcomes
1. Grasp the significance of audio in enhancing digital storytelling through emotion and immersion. 2. Recognise and differentiate various types of audio, including voiceovers, sound effects, and music. 3. Develop practical skills in editing audio using tools like AudioMass for tasks such as trimming, fading, and layering. 4. Understand the importance of sourcing audio resources ethically and responsibly. 5. Reflect on the creative process and evaluate personal learning experiences in audio production.	1. Explain the role of audio in enhancing digital storytelling with specific examples. 2. Identify and describe at least three types of audio used in digital media, such as voiceover, sound effects, and music. 3. Edit an audio clip using AudioMass by applying techniques like trimming, fading, and volume adjustment. 4. Source and import royalty-free audio resources responsibly, adhering to copyright guidelines. 5. Export a completed audio project in a suitable format, such as MP3, and verify its playback quality.

Lesson: Combining Media Elements

□ Intermediate	□ 80 mins	□ Student Quiz	□ Student Challenge

These teaching notes are designed to support educators in guiding students through a structured multimedia project. Encourage students to follow the step-by-step process, focusing on planning, editing, and integrating various media types using tools like Canva, AudioMass, and Clipchamp. Facilitate discussions on creativity and copyright adherence, ensuring students preview and refine their work regularly for a polished outcome.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the principles of integrating multiple media types into a cohesive multimedia project. 2. Develop skills in planning and structuring a multimedia narrative using storyboards. 3. Apply essential editing techniques to enhance media elements for professional output. 4. Demonstrate proficiency in using online tools for media creation and editing. 5. Create and export a polished multimedia piece that combines at least two media forms.	1. Plan a multimedia project by selecting a theme and creating a storyboard that integrates at least two media types. 2. Edit images using Canva, applying techniques like cropping and text overlays to support the project theme. 3. Process audio in AudioMass, using trimming and fading to enhance quality and fit the project narrative. 4. Combine media elements in Clipchamp, arranging and layering images, audio, and text for a cohesive output. 5. Export the final multimedia project in a suitable format, ensuring all elements are correctly integrated and playable.

Lesson: Audience and Purpose

□ Intermediate	□ 80 mins	□ Student Quiz	□ Student Challenge

These teaching notes are designed to support educators in delivering a lesson on key communication concepts. Begin by familiarising yourself with the step-by-step structure, focusing on audience identification and purpose definition. Encourage student engagement through reflective activities and creative tasks using Canva. Provide guidance on tailoring content and facilitate discussions to deepen understanding of design choices.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the concepts of audience and purpose in designing digital content. 2. Analyse how different audiences influence content tone, visuals, and format. 3. Evaluate the impact of purpose on design decisions and communication style. 4. Apply principles of audience and purpose to create tailored digital content. 5. Reflect on design choices and justify their effectiveness for specific goals.	1. Identify and describe at least three characteristics of a specific audience that influence digital content design. 2. Explain how a defined purpose (e.g., to inform, persuade, or entertain) shapes the tone, format, and style of content creation. 3. Create a short piece of digital content using Canva, tailored to a chosen audience and purpose, incorporating appropriate visuals and tone. 4. Justify design choices in a reflective paragraph, linking decisions to the selected audience and purpose. 5. Propose specific modifications to adapt the created content for a different audience, explaining the rationale behind each change.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Digital Media Literacy Short Course / Module: Intermediate Digital Media

Intermediate Digital Media

Lesson: Source Credibility

□ Intermediate	□ 80 mins	□ Student Quiz	□ Student Challenge

Prepare students to evaluate online information by introducing key concepts like the CRAAP Test, red flags, tone, and bias. Facilitate discussions on credibility using provided examples and activities. Encourage reflection and application through practical tasks like headline analysis and personal checklist creation to build critical thinking skills.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the importance of evaluating online information for trustworthiness. 2. Master the CRAAP Test criteria to assess source reliability systematically. 3. Identify red flags and biases that indicate unreliable or misleading content. 4. Apply critical thinking skills to distinguish credible sources from untrustworthy ones. 5. Develop personal strategies for responsible digital citizenship in online interactions.	1. Define source credibility and explain its importance in evaluating online information. 2. Apply the CRAAP Test criteria (Currency, Relevance, Authority, Accuracy, Purpose) to assess the reliability of given sources. 3. Identify red flags in online content, such as emotional language or lack of evidence, to determine potential unreliability. 4. Analyse tone and bias in written content to distinguish between neutral and one-sided information. 5. Create a personalised credibility checklist with 5-10 rules to evaluate online sources effectively.

Lesson: Spotting Bias

□ Intermediate	□ 80 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes

Lesson: Fact-Checking Skills

□ Intermediate	□ 80 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes

Lesson: Consumer Patterns Online

□ Intermediate	□ 80 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes

Lesson: Comparing Digital Formats

□ Intermediate	□ 80 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Digital Media Literacy Short Course / Module: Intermediate Digital Media

Intermediate Digital Media

Lesson: Introduction to Image Generation

□ Intermediate	□ 80 mins	□ Student Quiz	□ Student Challenge

These teaching notes are designed to support educators in delivering an engaging lesson on AI image generation. Begin by familiarising yourself with the step-by-step content, focusing on prompts, ethical considerations, and safe tools. Encourage student creativity through activities like crafting prompts and generating images. Facilitate discussions on real-world applications and ethical use, ensuring a balance of technical exploration and critical thinking.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Grasp the fundamental concepts and processes behind AI-generated imagery. 2. Understand the role of prompts in shaping AI image outputs. 3. Explore practical applications of AI image generation across various industries. 4. Develop skills in using safe, free AI tools to create original images. 5. Recognise ethical considerations and the importance of responsible AI use.	1. Explain the fundamental concept of AI-generated imagery and its key applications across various industries. 2. Construct effective prompts with specific details to guide AI in producing targeted image outputs. 3. Utilise safe, free online AI tools to generate original images based on textual descriptions. 4. Identify ethical considerations in AI image generation, focusing on issues like misinformation and respect for original creators. 5. Create a final AI-generated image project, incorporating detailed prompts and reflecting on the creative process.

Lesson: Prompt Engineering for Images

□ Intermediate	□ 80 mins	□ Student Quiz	□ Student Challenge

These teaching notes are designed to support educators in delivering a lesson on crafting effective prompts for AI image generation. Encourage students to explore safe tools like Craiyon and Canva AI, focusing on curiosity and creativity. Guide them through structured activities to build precise prompts, compare vague versus specific examples, and reflect on outcomes, ensuring a safe and engaging learning environment.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Master the art of crafting detailed prompts to guide AI in generating specific and creative images. 2. Understand the impact of word choice, tone, and structure on the quality and accuracy of AI-generated visuals. 3. Develop skills to break down and utilise key prompt components such as subject, setting, style, mood, and lighting. 4. Recognise the importance of specificity in prompts to achieve desired outcomes compared to vague instructions. 5. Foster a safe and ethical approach to using AI image generation tools while encouraging creativity and curiosity.	1. Construct detailed prompts for AI image generation using key components such as subject, setting, style, mood, and lighting. 2. Compare the outcomes of vague versus specific prompts to evaluate their impact on the accuracy and quality of generated images. 3. Experiment with word choices and tones to modify the atmosphere and emotional impact of AI-generated images. 4. Apply safe and ethical practices when using AI image generation tools, ensuring responsible and positive content creation. 5. Create and test multiple prompt variations on a chosen theme to analyse how different structures influence visual results.

Lesson: Ethical Considerations and Societal Impact

□ Intermediate	□ 80 mins	□ Student Quiz	□ Student Challenge

These teaching notes are designed to support educators in guiding students through a lesson on the ethical and societal dimensions of AI. Encourage critical thinking by facilitating discussions on bias, privacy, and misinformation. Prepare real-world examples to illustrate impacts, and ensure activities like creating a personal code of ethics are interactive and reflective for deeper engagement.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the ethical implications of AI in content generation. 2. Recognise the societal impacts of AI, both positive and negative. 3. Identify biases and privacy concerns in AI applications. 4. Develop critical thinking skills to address AI-related ethical dilemmas. 5. Formulate personal guidelines for responsible AI use.	1. Identify and explain at least two ethical issues related to AI-generated content, such as bias or privacy concerns. 2. Analyse real-world examples of AI's societal impact, highlighting both positive and negative effects. 3. Evaluate specific case studies to pinpoint ethical dilemmas in AI use and propose practical solutions. 4. Develop a personal code of ethics with at least five actionable statements for responsible AI application. 5. Demonstrate critical thinking by discussing strategies to mitigate risks like misinformation or copyright infringement in AI contexts.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Digital Media Literacy Short Course / Module: Advanced Digital Media

Advanced Digital Media

Lesson: Advanced Video Editing

□ Advanced	□ 200 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes

Lesson: Interactive Web Design

□ Advanced	□ 120 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes

Lesson: Digital Portfolio Creation

□ Advanced	□ 120 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes

Lesson: Social Media Campaign

□ Advanced	□ 160 mins	□ Student Quiz	□ Student Challenge

These teaching notes are designed to support educators in guiding students through a lesson on creating a mock social media campaign. Encourage students to explore real-world examples and analyse success factors. Facilitate discussions on theme selection and audience targeting. Provide time for creating sample posts and ensure access to tools like Canva for designing graphics.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the purpose and structure of effective social media campaigns. 2. Analyse key factors contributing to the success of digital campaigns. 3. Develop a mock campaign with a clear theme, goals, and target audience. 4. Create engaging content tailored to specific social media platforms. 5. Plan a content schedule and define metrics to evaluate campaign impact.	1. Explain the key elements that contribute to the success of a social media campaign, such as messaging and audience targeting. 2. Analyse real-world social media campaigns to identify factors behind their effectiveness. 3. Develop a mock social media campaign plan, including a defined theme, goals, and target audience profile. 4. Create three platform-specific sample posts with captions, visuals, and hashtags tailored to Instagram, X, and TikTok. 5. Construct a complete campaign schedule and define measurable success metrics to evaluate impact.

Lesson: Collaborative Project

□ Advanced	□ 200 mins

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Digital Media Literacy Short Course / Module: Advanced Digital Media

Advanced Digital Media

Lesson: Impact of Digital Media

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes

Lesson: Data Privacy Deep Dive

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes

Lesson: Bias in Algorithms

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes

Lesson: Reflecting on Digital Citizenship

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes

Lesson: Digital Ethics Debate

□ Advanced	□ 80 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Digital Media Literacy Short Course / Module: Advanced Digital Media

Advanced Digital Media

Lesson: How AI Works

□ Advanced	□ 40 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes

Lesson: Advanced Prompt Engineering

□ Advanced	□ 80 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes

Lesson: An Introduction to AI Models

□ Advanced	□ 20 mins	□ Student Quiz

Prepare to introduce students to AI models, explaining their function and various types. Discuss different learning methods such as supervised, unsupervised, and reinforcement learning. Explore the diverse applications of AI models, from speech recognition to autonomous vehicles. Discuss the limitations of AI models, including data quality and computational resources. Finally, delve into the ethics of AI models, discussing responsibility, privacy, transparency, and fairness.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand the concept and purpose of AI models. 2. Identify different types of AI models and their learning methods. 3. Recognise various applications of AI models in real-world scenarios. 4. Appreciate the limitations and challenges associated with AI models. 5. Reflect on the ethical considerations in the use of AI models.	1. Identify and describe the different types of AI models: Supervised Learning, Unsupervised Learning, and Reinforcement Learning. 2. Explain the various applications of AI models, including speech recognition, image recognition, natural language processing, recommendation systems, and autonomous vehicles. 3. Discuss the limitations of AI models, focusing on data quality, computational resources, transparency, privacy, and security. 4. Understand the ethical considerations related to AI models, including responsibility, privacy, transparency, and fairness. 5. Demonstrate a basic understanding of how AI models function, their uses, limitations, and ethical implications.

Lesson: Create an Image Model

□ Advanced	□ 20 mins	□ Student Quiz

Familiarise yourself with Google's Teachable Machine tool before the lesson. Ensure students understand the concept of machine learning and how it applies to image recognition. Encourage students to take clear images for their classes and emphasise the importance of quality over quantity. Guide them through the process of training, testing, and exporting their models. Reinforce the practical application of these skills in future projects.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Required equipment for this lesson:

- Webcam/camera

Learning Goals	Learning Outcomes
1. Understand and utilise Google's Teachable Machine to create an image model. 2. Create and define classes within an image model project. 3. Add and manage image samples to each class for effective model training. 4. Train, test, and refine the image model to ensure accurate gesture recognition. 5. Export and save the created image model for future use in projects.	1. Utilise Google's Teachable Machine to create an image model. 2. Create and categorise classes within an image model project. 3. Add and record images to each class using a webcam. 4. Train the image model using the added images and understand the process of machine learning. 5. Test the model's performance, make necessary adjustments, and export the model for future use.

Lesson: Create a Pose Model

□ Advanced	□ 20 mins	□ Student Quiz

Prepare to guide students through creating a pose model using Google's Teachable Machine. Familiarise yourself with the tool and the process of creating classes, adding images, and training the model. Be ready to troubleshoot any issues with webcam permissions or image quality. Ensure students understand the importance of testing their model and making necessary adjustments. Finally, assist them in exporting their model for future use in projects like an AI-powered space game.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Required equipment for this lesson:

- Webcam/camera

Learning Goals	Learning Outcomes
1. Develop an understanding of Google's Teachable Machine and its application in creating pose models. 2. Acquire skills to create and categorise classes within a pose model. 3. Learn to add and manage image samples for each class to train the model. 4. Gain proficiency in training and testing the model for different poses. 5. Master the process of exporting the model for future use in other projects.	1. Operate Google's Teachable Machine to create a pose model. 2. Define and create classes for the pose model. 3. Add and categorise images into the respective classes: Tilt Left, Tilt Right, and No Tilt. 4. Train the pose model using the categorised images and test its performance. 5. Export the created pose model and obtain a shareable link for future use.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Digital Media Literacy Short Course / Module: Digital Media Literacy CBA

Digital Media Literacy CBA

Lesson: Footprint Analysis

Intermediate	120 mins

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes

Lesson: Creating the Presentation

Intermediate	240 mins

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes

Lesson: Presenting and Reflecting

Intermediate	120 mins

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes

Digital Media Literacy CBA

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Digital Media Literacy Short Course / Module: Digital Media Literacy CBA

Digital Media Literacy CBA

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Digital Media Literacy Short Course / Module: Digital Media Literacy CBA

Digital Media Literacy CBA

Lesson: Introduction & Proposal

□ Advanced	□ 80 mins

These teachers, guide students through initiating their Classroom-Based Assessment in Digital Media Literacy. Start by discussing the project overview, encouraging personal topic selection. Review the Features of Quality criteria to set clear goals. Facilitate analysis and discussion of sample projects, then support brainstorming and refining individual plans with reflective exercises.

Learning Goals	Learning Outcomes
1. Understand the purpose and scope of the Classroom-Based Assessment in digital media literacy. 2. Identify personal interests and relevant topics for a self-directed digital media project. 3. Familiarise with the Features of Quality criteria to guide project planning and execution. 4. Analyse sample projects to apply assessment criteria and recognise quality standards. 5. Develop a structured project plan incorporating creativity, research, and ethical considerations.	1. Identify a personal topic of interest related to digital media literacy and justify its relevance to real-world issues. 2. Analyse the Features of Quality criteria to set specific achievement goals for the Classroom-Based Assessment project. 3. Evaluate sample projects against assessment criteria to identify strengths and areas for improvement applicable to personal work. 4. Develop a detailed project plan incorporating chosen topic, digital tools, target audience, and alignment with Features of Quality. 5. Reflect on personal goals and strategies to achieve a targeted level of success in the project.

Lesson: Research & Source Evaluation

□ Advanced	□ 80 mins

Learning Goals	Learning Outcomes

Lesson: Creation Phase

□ Advanced	□ 240 mins

Learning Goals	Learning Outcomes

Lesson: Peer Feedback and Refinement

□ Advanced	□ 80 mins

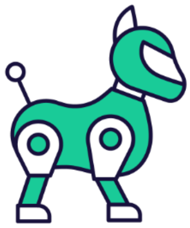
Learning Goals	Learning Outcomes

Lesson: Finalising & Reflection

□ Advanced	□ 120 mins

Learning Goals	Learning Outcomes

Robotics & Engineering Short Course



This short course introduces students to the fascinating world of robotics and engineering through engaging lessons and hands-on projects. As a teacher, focus on sparking curiosity by using real-world examples and encouraging experimentation. Ensure access to necessary tools like Microbits and kits, and guide students through coding and assembly with patience, fostering creativity and problem-solving skills.

Duration	Equipment
Classroom hours ~68	Students can use any of these devices: • Chromebook/Laptop/PC • Microbit Required Equipment: • LED Strip with crocodile clips • Microbit • Move Motor Car • Move Motor Klaw • Phillips Screwdriver • Raspberry Pi PICO W • Smart Home Kit • Traffic Lights Kit • USB Cable
Learning Goals	**Learning Outcomes**
1. Understand the fundamental concepts and components of robotics, including their design, functionality, and applications across various industries. 2. Trace the historical development of robotics and evaluate its societal impact and future potential. 3. Develop practical skills in programming and building robotic systems using tools like Microbits and Raspberry Pi Pico. 4. Explore innovative applications of robotics in solving real-world problems, from healthcare to space exploration. 5. Analyse ethical considerations and challenges associated with robotics and automation in modern society.	1. Define and explain the fundamental concepts of robotics, including the components and functions of robots such as sensors, motors, and controllers. 2. Trace the historical development of robotics from early mechanical devices to modern AI-driven systems, identifying key milestones and societal impacts. 3. Analyse potential future trends in robotics, evaluating the role of AI, human-robot collaboration, and ethical considerations in shaping these advancements. 4. Demonstrate practical skills by programming a Microbit to perform specific tasks, such as creating a step counter or controlling LED strips for interactive projects. 5. Apply knowledge of robot applications by designing and testing a functional robotic system, such as a line-following car or traffic light simulation, using appropriate hardware and software tools.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Robotics & Engineering Short Course / Module: Robotics & Engineering Basics

Robotics & Engineering Basics

Lesson: What is a Robot?

□ Beginner	□ 20 mins	□ Student Quiz	□ Student Challenge

Begin with an introduction to robots, explaining their autonomous nature and components like sensors and motors. Discuss the variety of robot appearances, from humanoid to animal-inspired and task-specific designs. Highlight the versatility of robots in performing tasks, from household chores to industrial assembly and exploration. Explain the basic functioning of robots, focusing on sensors, motors, and controllers. Illustrate the wide application of robots in settings like factories, hospitals, and hazardous environments. Conclude by discussing the importance of robots in performing dangerous tasks, increasing efficiency, and working alongside humans.

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

- Chromebook/Laptop/PC

Learning Goals	Learning Outcomes
1. Understand the definition and basic components of robots. 2. Recognise the variety of robot designs and their purposes. 3. Identify the range of tasks robots can perform. 4. Appreciate the diverse applications of robots across different environments. 5. Evaluate the importance and benefits of robots in society.	1. Identify and describe different types of robots based on their appearance and design. 2. List various tasks robots can perform across different settings. 3. Explain the function of key robot components: sensors, motors, and controllers. 4. Name five environments where robots are used and describe their roles in each. 5. Discuss the benefits of robots in performing hazardous, challenging, or repetitive tasks.

Lesson: History of robotics

□ Beginner	□ 20 mins	□ Student Quiz	□ Student Challenge

Begin with an engaging introduction to the history of robotics, highlighting its origins from ancient mechanical devices to the Industrial Revolution. Discuss early mechanical devices like water clocks and automata, focusing on their functionality and influence on modern robotics. Explore the Industrial Revolution's role in advancing automation and machine development. Introduce early robots from the 1950s, their industrial applications, and the origin of the term 'robot'. Cover advancements in robotics, including AI and sensors, and their diverse applications today. Discuss the impact of robotics on efficiency, productivity, and ethical considerations. Conclude with a look at the future of robotics, discussing potential societal implications and benefits. Encourage students to research and reflect on each step's topics.

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

- Chromebook/Laptop/PC

Learning Goals	Learning Outcomes
1. Understand the evolution of robotics from ancient mechanical devices to modern technology. 2. Recognise the impact of the Industrial Revolution on the development of robotics. 3. Identify key milestones in the history of robotics, including the introduction of the term 'robot' and early robot designs. 4. Analyse the current applications and advancements in robotics across various fields. 5. Evaluate the societal impact and ethical considerations of robotics, including future possibilities and implications.	1. Identify and describe early mechanical devices from ancient Greece and Rome. 2. Explain the role of the Industrial Revolution in the development of robotics. 3. Summarise the characteristics and uses of early robots from the 1950s. 4. Discuss the impact of modern robotics on efficiency and productivity in various fields. 5. Analyse the ethical considerations and future implications of advanced robotics.

Lesson: Future of robotics

□ Beginner	□ 20 mins	□ Student Quiz	□ Student Challenge

Begin with an engaging introduction to the future of robotics, highlighting global efforts and innovations. Discuss AI's role in robotics, encouraging students to research examples and consider real-world applications. Explore human-robot collaboration by having students imagine a robot assisting in daily tasks and discuss design and safety features. Introduce biologically inspired robotics with examples of nature-inspired designs. Cover robots in space exploration, focusing on their role in harsh environments. Address ethical considerations, prompting discussion on job displacement, privacy, and misuse. Conclude with a reflection on how robotics advancements could impact society, encouraging students to share their thoughts on future opportunities and challenges.

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

- Chromebook/Laptop/PC

Learning Goals	Learning Outcomes
1. Understand the role of AI in advancing robotics capabilities. 2. Explore the potential of human-robot collaboration in daily tasks. 3. Recognise the influence of biological systems on robot design. 4. Appreciate the significance of robots in space exploration. 5. Consider ethical implications of increasing robot integration into society.	1. Identify examples of AI in robotics and explain their potential real-world applications. 2. Design a robot that collaborates with humans, specifying its functions and safety features. 3. Describe how biologically inspired designs enhance robot functionality. 4. Explain the role of robots in space exploration and their specific tasks. 5. Discuss ethical issues related to advanced robotics and propose solutions.

Lesson: Robot anatomy

□ Beginner	□ 30 mins	□ Student Quiz	□ Student Challenge

Prepare to introduce robot anatomy, comparing it to human anatomy for relatability. Discuss key components like sensors, actuators, and controllers, using analogies to clarify their functions. Explore various robot body designs and their task-specific adaptations. Explain different locomotion types, using examples to illustrate each. Detail end effectors, their types, and purposes. Discuss power supply options, their advantages, and limitations. Conclude by reinforcing the importance of each component in robot functionality and design. Encourage students to think about designing their own robots.

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

- Chromebook/Laptop/PC

Learning Goals	Learning Outcomes
1. Understand the basic components and structure of robots. 2. Recognise different types of robot locomotion and their applications. 3. Identify various end effectors and their specific functions. 4. Comprehend the role and types of power supplies in robots. 5. Apply knowledge of robot anatomy to design and build robots for specific tasks.	1. Identify and describe the main components of a robot, including sensors, actuators, and controllers. 2. Explain the purpose and function of a robot's body structure in relation to its tasks. 3. Compare and contrast different types of robot locomotion, such as wheeled, legged, tracked, flying, and swimming. 4. Select appropriate end effectors for specific robotic tasks, understanding their functions and applications. 5. Analyse the suitability of different power sources for robots based on their intended use and environment.

Lesson: Robot applications

□ Beginner	□ 20 mins	□ Student Quiz	□ Student Challenge

Begin with an engaging introduction to robots, highlighting their presence in various sectors. Discuss manufacturing and industrial uses, focusing on automation and safety. Explore healthcare applications, detailing specific types like robotic surgery and rehabilitation. Cover space exploration, mentioning Mars rovers and robotic arms. Explain service industry roles, such as in hotels. Address military applications, including ethical considerations. Conclude by inspiring students about future possibilities in robotics.

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

- Chromebook/Laptop/PC

Learning Goals	Learning Outcomes
1. Understand the diverse applications of robots across various industries. 2. Recognise the impact of robots on efficiency and safety in manufacturing and industrial settings. 3. Explore the role of robots in enhancing healthcare services and patient outcomes. 4. Appreciate the significance of robots in space exploration and their capabilities in harsh environments. 5. Consider the ethical implications of using robots in military and service applications.	1. Identify and describe various applications of robots in manufacturing and industry. 2. Explain the role of robots in healthcare, including specific examples like robotic surgery and rehabilitation. 3. Discuss how robots are used in space exploration, detailing their functions and benefits. 4. Outline the use of robots in service industries, highlighting their efficiency and capabilities. 5. Analyse the applications and ethical considerations of robots in military contexts.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Robotics & Engineering Short Course / Module: Robotics & Engineering Basics

Robotics & Engineering Basics

Lesson: Exploring Microbits

□ Beginner	□ 60 mins	□ Student Quiz

Prepare to introduce students to the world of microbits, a pocket-sized programmable computer. The lesson will involve creating a new project on the MakeCode for microbit website, familiarising with the project editor, and writing code to display numbers, names, and icons. Students will also learn to delete code, connect their microbits to their computers, and program their microbits to play music. The lesson concludes with an exploration phase where students can experiment with different blocks from the toolbox.

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. Create a new project using the MakeCode for microbit website. 3. Use the Project Editor to write and simulate code. 4. Program the microbit to display numbers and text on its LED grid. 5. Program the microbit to respond to button presses with specific actions.	1. Identify the functions and capabilities of a microbit. 2. Create a new project on the MakeCode for microbit website. 3. Understand the layout and functions of the Project Editor. 4. Write and execute code to display numbers and names on the microbit. 5. Program the microbit to respond to button presses with specific displays. 6. Connect and download code to an actual microbit device. 7. Compose and program a melody to play on the microbit. 8. Explore and experiment with different coding blocks and functions.

Lesson: Microbit Step Counter

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through creating a Microbit step counter. They'll start a new project on makecode.microbit.org, create and set up a 'steps' variable, and use the accelerometer to detect steps. They'll write code to display the step count and send it to their Microbit. After connecting a power source, they'll secure the Microbit to their person and start walking. They'll adjust the code to count every step and resend the updated code to their Microbit. Caution them to be careful while walking with the Microbit.

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 a basic understanding of Microbit programming and project creation. 2. Learn to create and set up variables in Microbit. 3. Understand the use of accelerometer sensor in Microbit for step detection. 4. Gain skills to display data on Microbit using LEDs. 5. Learn to modify and resend code to Microbit for improved functionality.	1. Develop a new Microbit project using the makecode.microbit.org website. 2. Create and set up a 'steps' variable to record the number of steps taken. 3. Utilise the accelerometer sensor in Microbits to detect and record steps. 4. Display the recorded number of steps on the Microbit using its LEDs. 5. Modify the code to accurately count every step taken, and resend the updated code to the Microbit.

Lesson: Reaction Timer

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students in creating a 'Reaction Timer' project using Micro:bit. They'll start by setting up a new project, then create a welcome message and a countdown. Next, they'll add a random delay to make the game unpredictable. They'll create variables to store time stamps, and finally, record the player's reaction time. Familiarise yourself with the code snippets provided.

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 in creating and managing a new project on the Micro:bit platform. 2. Acquire knowledge on how to create and display messages using code. 3. Understand and apply the concept of countdowns and delays in programming. 4. Learn to create and utilise variables for storing time stamps. 5. Gain proficiency in recording and displaying user interactions in real-time.	1. Develop a new project using the Micro:bit website. 2. Construct a welcome message to display upon powering on the Microbit. 3. Create a countdown sequence with visual cues using code. 4. Implement a random delay function in the game for unpredictability. 5. Create and utilise variables to store time stamps. 6. Record and display player reaction time upon button press.

Lesson: Microbit Fruit and Veg Piano

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to demonstrate the conductivity of the human body and various fruits and vegetables using a Microbit. Gather a Microbit, 4 crocodile clips, and 4 pieces of fruit or vegetables. Familiarise yourself with the Microbit programming interface and the specific code for programming Pins 0, 1, and 2. Ensure you understand how to connect the crocodile clips and test the circuits. Be ready to guide students in connecting the fruit and vegetables to create a musical instrument.

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 concept of electrical conductivity using the human body and various fruits and vegetables. 2. Identify and utilise the components of a Microbit, including its pins and GND. 3. Create and modify a Microbit project using the makecode.microbit.org platform. 4. Program Microbit pins to play different musical notes and display different icons. 5. Test and troubleshoot a simple electrical circuit using a Microbit, crocodile clips, and conductive materials.	1. Identify and gather necessary materials for creating an electrical circuit with a Microbit and fruit or vegetables. 2. Create a new project on the makecode.microbit.org website. 3. Program Pins 0, 1, and 2 on the Microbit to play different notes and display different icons when pressed. 4. Connect crocodile clips to Pins 0, 1, 2 and GND on the Microbit and test the circuit. 5. Attach fruit or vegetables to the crocodile clips and demonstrate the ability to play different notes by touching and releasing each piece.

Lesson: Designing a Microbits Weather Station

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare for this lesson by familiarising yourself with the MakeCode for Microbit platform and the coding language used. Understand the purpose of variables and how they can be initialised and manipulated. Be prepared to guide students through the process of creating a new project, configuring buttons and sensors, creating a 'forever' loop, and testing their program. Encourage reflection on the learning process and potential applications of the skills learned.

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 how to create a new project on MakeCode for Microbit. 2. Learn to declare and initialise variables in a Microbit project. 3. Gain skills in configuring Microbit buttons and sound detection. 4. Develop the ability to create a 'forever' loop and display sensor data. 5. Reflect on the learning process and consider potential applications of Microbit sensors.	1. Develop a new project using MakeCode for Microbit. 2. Declare and initialise variables 'mode' and 'reading' for sensor data display and storage. 3. Configure Button A to set 'mode' to 1 when pressed. 4. Configure Button B to set 'mode' to 2 when pressed. 5. Configure Buttons A and B together to set 'mode' to 3 when pressed simultaneously. 6. Configure the Microbit to switch to mode 4 when a loud sound is detected. 7. Create a 'forever' loop to check the value of 'mode' and display the relevant sensor data. 8. Test the program using a physical Microbit or the simulator on the MakeCode website. 9. Reflect on the learning process, understanding how the different sensors on the Microbit work and potential other projects.

Lesson: Microbit Compass and Thermometer

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students in creating a Microbit project that utilises the compass and temperature sensor. They will learn to create and set variables, program buttons, and use 'if then else' blocks. The lesson involves coding the Microbit to display cardinal directions based on its orientation and temperature readings. Students will also test their code using a simulator before sending it to their Microbit. Ensure familiarity with the makecode.com platform and basic coding concepts.

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 compass and temperature sensor features of the Microbit. 2. Develop proficiency in creating and setting variables in a Microbit project. 3. Apply conditional logic to program Microbit buttons for specific functions. 4. Test and debug code using the simulator before transferring to the Microbit. 5. Interpret and display data from the Microbit's sensors in a user-friendly format.	1. Develop a new Microbit project using makecode.com. 2. Create and set a 'direction' variable to store compass readings. 3. Program the A button to display compass direction (N, S, E, W) based on 'direction' variable. 4. Program the B button to display the current temperature reading. 5. Test and debug the code using the simulator and then deploy it to the Microbit.

Lesson: Microbit Pet

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

In this lesson, students will transform their Microbits into interactive pets. They will use emoji icons and sounds to make the Microbits seem lifelike, programming them to respond to different actions such as shaking, touching, and flipping. Students will create functions for different states of the pet, like happy, sad, hungry, bored, and asleep. They will also learn to use the Microbit's sensors to detect these actions. The lesson involves coding in the Microbit's online editor, testing the code in a simulator, and finally downloading it onto their Microbits.

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 in creating and using functions in Microbit programming. 2. Understand and apply the use of different sensors on the Microbit device. 3. Gain knowledge in programming interactive responses using sound and visual cues. 4. Learn to use random time intervals in programming for unpredictable outcomes. 5. Enhance problem-solving skills by debugging and testing code in a simulator and on a physical device.	1. Program Microbit to display different emoji icons and sounds to simulate pet behaviours. 2. Create a new Microbit project using the provided website. 3. Develop functions such as 'happy', 'feedme', and 'play' to control pet behaviours. 4. Implement gesture controls to interact with the Microbit pet, such as shaking, flipping, and touching the logo. 5. Test the programmed Microbit pet in the simulator and download the code onto a physical Microbit.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Robotics & Engineering Short Course / Module: Robotics & Engineering Basics

Robotics & Engineering Basics

Lesson: Tilt Angle

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

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

Lesson: Acceleration

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

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

Lesson: Noise Level

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

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

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Robotics & Engineering Short Course / Module: Robotics & Engineering Basics

Robotics & Engineering Basics

Lesson: Microbit LED Strip

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through programming a 30 LED strip using Microbits. Ensure understanding of creating a new Microbit project and adding the neopixel extension. Facilitate the setup of the LED strip and programming it to turn red. Assist with downloading the project onto the Microbit. Encourage creativity when programming the strip to show a rainbow of colours and rotating the rainbow. Finally, encourage exploration of other code blocks in the Neopixel toolbox.

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- LED Strip with crocodile clips

Learning Goals	Learning Outcomes
1. Understand and apply the process of programming a strip of LEDs using Microbits. 2. Develop skills in creating a new Microbit project and adding the necessary extensions. 3. Gain proficiency in setting up and programming the LED strip to display various colours. 4. Learn to download and implement the project on Microbits, observing the effects on the LED strip. 5. Explore and experiment with different code blocks in the Neopixel toolbox for creative lighting effects.	1. Program a strip of 30 LEDs to light up in different ways using Microbits. 2. Create a new Microbit project and add the neopixel extension. 3. Set up the LED strip and interact with it using a variable. 4. Program the A button on the Microbit to turn all the LEDs red. 5. Program the LED strip to show a rainbow of colours when the Microbit turns on.

Lesson: LED Strip Clapper

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

In this lesson, students will create an LED Strip Clapper using a Microbit project. They will add the neopixel extension, set up the LED strip, and create an 'on' variable. The lesson will guide them to detect a clap, turning the LED strip on and off accordingly. They will download their code onto their microbit, connect it to the LED strip, and explore further improvements. Familiarity with Microbit and basic coding is beneficial.

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- LED Strip with crocodile clips

Learning Goals	Learning Outcomes
1. Develop skills in creating and managing a new Microbit project. 2. Understand and apply the neopixel extension for programming an LED strip. 3. Learn to set up and interact with the LED strip using variables. 4. Gain knowledge on creating and manipulating variables to control the state of the LED strip. 5. Develop the ability to detect sound inputs and use them to trigger changes in the LED strip's state.	1. Develop a new Microbit project using makecode.microbit.org. 2. Integrate the neopixel extension into the project for LED strip programming. 3. Establish a variable for the LED strip and set its value to 30. 4. Create an 'on' variable to control the LED strip's state. 5. Implement a sound detection feature to trigger the LED strip's state change.

Lesson: Microbit LED Strip Thermometer

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare for this lesson by familiarising yourself with the Microbit project platform and the neopixel extension. Understand how to set up the LED strip and how to program the A button to display temperature. Be ready to guide students in lighting up the LED lights according to temperature readings and downloading their projects onto their Microbits. Ensure you know how to correctly connect the LED strip to the Microbit.

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- LED Strip with crocodile clips

Learning Goals	Learning Outcomes
1. Develop skills to create and manage a new Microbit project. 2. Understand and apply the neopixel extension to program the LED strip. 3. Gain knowledge on setting up the LED strip and displaying temperature on the Microbit screen. 4. Learn to light up the LED lights on the strip according to the temperature readings. 5. Acquire practical skills in downloading the project, connecting the LED strip to the Microbit, and testing the functionality.	1. Create a new Microbit project using makecode.microbit.org. 2. Add the neopixel extension to the project for LED strip programming. 3. Set up the LED strip in the project with a value of 30, representing the 30 LEDs on the strip. 4. Program the A button to display the temperature on the Microbit screen. 5. Display the temperature by lighting up the LED lights on the strip, with the number of lights corresponding to the temperature reading. 6. Download the project and transfer it to the Microbit. 7. Connect the LED strip to the Microbit using the specified pin connections and power it using a USB cable.

Lesson: Shooting Stars

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students in creating a Microbit project, adding the neopixel extension, and setting up the LED strip. Facilitate the creation of a 'star' that lights up with a loud sound, and ensure students can test this on their LED strip. Assist students in making the 'star' shoot along the strip and adding random colours. Finally, ensure students can download and test their code, encouraging them to create multiple shooting stars.

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- LED Strip with crocodile clips

Learning Goals	Learning Outcomes
1. Develop skills in creating and managing a new Microbit project. 2. Understand and apply the neopixel extension to program the LED strip. 3. Gain proficiency in setting up and programming the LED strip using code blocks. 4. Learn to utilise the microphone in the microbit to detect sound and trigger LED actions. 5. Acquire knowledge on how to test and debug the project on the LED strip. 6. Master the concept of pixel shifting to create the illusion of moving light. 7. Experiment with random colour generation for the LED strip. 8. Learn to download and implement the code onto the microbit for real-world testing.	1. Create and manage a new Microbit project. 2. Integrate the neopixel extension into the project. 3. Set up and programme the LED strip using the provided code. 4. Develop a function to light up the first LED on the strip white when a loud sound is detected. 5. Test the function on the LED strip and ensure it works as expected. 6. Implement a function to make the 'star' shoot along the strip. 7. Enhance the function to display stars in random colours. 8. Download and test the final code on the microbit, ensuring different colour 'stars' shoot along the strip when a loud noise is made.

Lesson: LED Flags

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students in creating LED flags using a Microbit project. They will need to understand how to add the neopixel extension and set up the LED strip. Facilitate as they create bicolor and tricolor flags, using the example of Malta and Ireland respectively. Encourage creativity and problem-solving skills for the challenge of representing the American flag.

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- LED Strip with crocodile clips

Learning Goals	Learning Outcomes
1. Understand and apply the concept of bicolor and tricolor flags using LED strips. 2. Create and manage a new Microbit project effectively. 3. Utilise the neopixel extension to program the LED strip. 4. Develop skills to set up and interact with the LED strip using code. 5. Apply coding skills to create complex patterns, such as the American flag, on the LED strip.	1. Construct bicolor and tricolor flags using LED strips. 2. Utilise the neopixel extension to program the LED strip. 3. Set up and interact with the LED strip using a variable. 4. Apply the concept of ranges to light up specific sections of the LED strip. 5. Code the LED strip to represent complex flag designs, such as the American flag.

Lesson: LED Stacking

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare for this lesson by familiarising yourself with the Microbit project platform and the neopixel extension. Understand how to set up an LED strip and create variables to store the strip and the amount of LEDs. Be ready to guide students in creating a function to show the LED stack, and programming buttons to increase and decrease the stack. Ensure students know how to download their code and connect their LED strip to their microbit.

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- LED Strip with crocodile clips

Learning Goals	Learning Outcomes
1. Develop skills in creating and managing a new Microbit project. 2. Understand and apply the neopixel extension for LED programming. 3. Learn to set up and interact with the LED strip using variables. 4. Develop competency in creating and using functions to control LED display. 5. Gain experience in programming button controls to manipulate LED stack size.	1. Create a new Microbit project using makecode.microbit.org. 2. Add the neopixel extension to the project for LED strip programming. 3. Set up the LED strip with a variable storing the strip, set to a value of 30. 4. Create an 'amount' variable to store the number of LEDs in the stack. 5. Develop a 'showStack' function to display the stack of lit LEDs. 6. Create a range of LEDs on the strip to light up, using the 'amount' variable, and call the 'showStack' function from the 'on start' block. 7. Program button A to increase the LED stack by adding 1 to the 'amount' variable and calling the 'showStack' function. 8. Program button B to decrease the LED stack by subtracting 1 from the 'amount' variable and calling the 'showStack' function. 9. Download the code onto a microbit, connect the LED strip using crocodile clips, and test the LED stack's increase and decrease functions with buttons A and B.

Lesson: LED Strip Precision Game

□ Advanced	□ 80 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students through creating an interactive LED strip game using a Microbit project. Familiarise yourself with the neopixel extension and the process of setting up the LED strip. Understand the purpose of the four variables: 'target', 'position', 'delay', and 'increment'. Be ready to explain how to set up the level, create a refresh function, and make the blue light move. Prepare to guide students through the steps of going back to the start, hitting the target, and handling a missed target. Finally, ensure you can assist students in downloading their code and playing the game.

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- LED Strip with crocodile clips

Learning Goals	Learning Outcomes
1. Understand and apply the concept of LED strip programming using Microbit. 2. Develop skills in creating and manipulating variables in a coding project. 3. Learn to create and use functions for specific tasks within a coding project. 4. Gain proficiency in using conditional statements to control game outcomes. 5. Develop the ability to download and test code on a physical device.	1. Program an LED strip to light up specific LEDs in response to user input. 2. Create and manipulate variables to control game mechanics in a Microbit project. 3. Implement the neopixel extension to interact with an LED strip. 4. Design a function to refresh LED lights based on variable values. 5. Download and test the code on a physical Microbit device.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Robotics & Engineering Short Course / Module: Intermediate Robotics & Engineering

Intermediate Robotics & Engineering

Lesson: Microbit Light Clapper

□ Beginner	□ 40 mins	□ Student Quiz	□ Student Challenge

Prepare for the 'Microbit Light Clapper' lesson by familiarising yourself with the makecode.microbit.org website. Understand the process of creating a new project, setting up variables, and using sound thresholds. Be ready to guide students in writing code to detect claps and control LED lights. Ensure you can troubleshoot issues and explain how to test the code in the simulator and on the Microbit.

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 concept of variables in coding. 2. Set and utilise sound thresholds for input detection. 3. Implement conditional statements (if-then-else) to control LED light responses. 4. Test and debug code in a simulator environment. 5. Transfer and apply code to a physical Microbit device.	1. Develop a new project using makecode.microbit.org. 2. Create and utilise a variable to control the LED lights on the Microbit. 3. Set a sound threshold for detecting claps using the Microbit's microphone. 4. Implement code to detect a clap based on the set sound threshold. 5. Use an 'if then else' block to control the LED lights based on the clap detection.

Lesson: Microbit Sounds

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare by ensuring access to MakeCode and micro:bit devices. Begin with an introduction to sound programming. Guide students through creating a new project named 'Music Maker'. Demonstrate playing melodies with the 'A' button, tones with the 'B' button, and sound effects with 'A+B'. Encourage experimentation with different musical elements. Conclude by testing the music on simulators or physical devices.

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 basics of programming music and sound on a microbit. 2. Create and modify melodies using different notes and tempos. 3. Program and experiment with playing individual tones. 4. Design and adjust sound effects using various waveforms and settings. 5. Test and refine music and sound projects on a microbit.	1. Create a new microbit project named 'Music Maker'. 2. Program the 'A' button to play a custom melody using the 'music.play' function. 3. Program the 'B' button to play a sequence of tones using the 'music.playTone' function. 4. Program the A+B buttons to play a custom sound effect using the 'music.playSoundEffect' function. 5. Test the music by pressing buttons on the simulator or a physical Microbit.

Lesson: Microbit Paddle Ball

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

In this lesson, students will create a Microbit Paddle Ball game using Scratch. They will learn to create a new project, add and position sprites, and make the ball bounce around the screen. They will also connect a Microbit to control the paddle, make the ball bounce off the paddle, add a backdrop, and create a game over line. The lesson concludes with programming the game over functionality and discussing potential improvements to the game. Teachers should familiarise themselves with Scratch and Microbit prior to the lesson.

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 in creating and managing a new Scratch project. 2. Understand and apply the concept of adding and positioning sprites in Scratch. 3. Gain proficiency in coding for sprite movement, interaction, and control using Scratch blocks. 4. Learn to integrate and use a Microbit with Scratch for real-time control of sprites. 5. Enhance critical thinking and problem-solving skills by identifying potential improvements to the game.	1. Develop a new Scratch project and remove the default cat sprite. 2. Add and position the 'Paddle' sprite from the sprite library. 3. Add the 'Soccer Ball' sprite from the sprite library. 4. Set the X and Y coordinates to position the ball at the top center of the screen. 5. Code the ball to move around the screen and bounce off the edges. 6. Connect and configure a Microbit to the Scratch project. 7. Code the paddle to move left and right by tilting the Microbit. 8. Program the ball to bounce off the paddle when it touches it. 9. Add the 'Stars' backdrop from the backdrop library. 10. Draw a red line at the bottom of the screen for the game over line. 11. Code the game to end when the ball touches the red game over line. 12. Propose improvements to the game by adding to or changing the existing code.

Lesson: Microbit Seismic and Meteorological Station

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

In this step-by-step lesson, you'll programme four Microbits to create a Seismic and Meteorological Station. Each Microbit will monitor and display different data: temperature, light levels, and simulated seismic activity. You'll then test your station, and explore ways to enhance it, such as recording historical data or setting specific alert conditions.

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 role of each Microbit in the Seismic and Meteorological Station and how they communicate with each other. 2. Develop skills in coding and programming Microbits to monitor and display temperature, light levels, and seismic activity. 3. Test and troubleshoot the coded Microbits to ensure they function as intended in the Seismic and Meteorological Station. 4. Apply critical thinking to improve the functionality of the Seismic and Meteorological Station. 5. Develop an understanding of how meteorological and seismic data can be collected, displayed, and used in real-world applications.	1. Program four separate Microbits to perform unique roles in a Seismic and Meteorological Station. 2. Code the 'Seismic and Meteorological Station' Microbit to monitor and wirelessly broadcast temperature, light levels, and 'seismic' activity data. 3. Code the 'Temperature Display' Microbit to receive and display the temperature data from the 'Seismic and Meteorological Station' Microbit. 4. Code the 'Day/Night Indicator' Microbit to receive light level data and display an indication of whether it's day or night. 5. Code the 'Seismic Alert' Microbit to receive 'seismic' activity data and display an alert if seismic activity is detected.

Lesson: Microbit Voting System

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students in creating a microbit voting system. They'll create two projects on the MakeCode Microbit website, one for voting microbits and another for a central microbit. They'll program the A and B buttons to cast votes, set up the central microbit to receive votes and reset the system, and display the vote results. They'll also enhance the system with a security feature. Ensure students understand the coding involved and the importance of testing their system.

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 a microbit voting system with separate voting and central microbits. 2. Programme the voting microbits to cast a single 'Yes' or 'No' vote. 3. Configure the central microbit to receive votes, count them, and reset the voting system. 4. Implement a reset function on the voting microbits to allow for multiple rounds of voting. 5. Enhance the voting system by adding a security feature to ensure the integrity of the votes.	1. Develop two separate projects on the MakeCode Microbit website for voting and central microbits. 2. Programme the A and B buttons on the microbit to cast a single 'Yes' or 'No' vote. 3. Set up the central microbit to receive votes, count them, and reset the voting system when needed. 4. Configure the individual voting microbits to receive the 'Reset' signal from the central microbit. 5. Display the vote results on the central microbit.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Robotics & Engineering Short Course / Module: Intermediate Robotics & Engineering

Intermediate Robotics & Engineering

Lesson: Build your Traffic Lights

□ Beginner	□ 10 mins	□ Student Quiz

Ensure students have all necessary materials, including the Microbit Traffic Lights Kit, a Microbit, and a Phillips head screwdriver. Guide them through opening the package and assembling the stand. Assist them in correctly positioning the Microbit on the traffic lights, ensuring they align the holes correctly. Supervise as they use the screwdriver to secure the Microbit. Celebrate their accomplishment once completed.

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- Traffic Lights Kit
- Phillips Screwdriver

Learning Goals	Learning Outcomes
1. Identify and gather necessary components for the Microbit Traffic Lights Kit. 2. Understand and execute the process of unpacking and preparing the kit. 3. Develop skills in assembling the stand for the traffic lights. 4. Apply knowledge of Microbit to correctly align and attach it to the traffic lights. 5. Demonstrate the ability to follow step-by-step instructions to complete a technical task.	1. Identify and gather necessary components for the Microbit Traffic Lights Kit. 2. Unpack and organise the Microbit Traffic Lights package contents. 3. Assemble the stand from the provided parts in the kit. 4. Align and attach the Microbit to the traffic lights using the correct hole configuration. 5. Successfully complete the assembly of the Microbit Traffic Lights.

Lesson: Microbit Traffic Lights

□ Beginner	□ 40 mins	□ Student Quiz	□ Student Challenge

In this lesson, students will create a new Microbit project on makecode.com, add the Stopbit extension, and test all the lights. They will learn about sequences in coding and apply this knowledge to program a traffic light sequence using on/off and state methods. Students will need to check the correct display of lights in each sequence.

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- Traffic Lights Kit

Learning Goals	Learning Outcomes
1. Understand and apply the process of creating a new Microbit project. 2. Learn to add and utilise the Stopbit extension for programming traffic lights kit. 3. Gain skills in testing and troubleshooting the functionality of the lights. 4. Comprehend the concept of 'sequence' in coding and apply it to program traffic lights. 5. Develop proficiency in programming the sequence of traffic lights using on/off and state methods.	1. Create and manage a new Microbit project on makecode.com. 2. Add and utilise the "stopbit" extension to the Microbit project. 3. Test and troubleshoot the functionality of each light on the Microbit. 4. Understand and apply the concept of 'sequence' in coding to program traffic lights. 5. Program the sequence of traffic lights using on/off and state methods.

Lesson: Traffic Light Reaction Game

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to guide students in creating a traffic light reaction game using a micro:bit and STOP:bit Traffic Lights. Ensure students understand how to attach the traffic lights to the micro:bit, create a new project on MakeCode, and add the "stopbit" extension. Explain the purpose of the 'startTime', 'endTime', and 'reactionTime' variables. Walk them through the process of setting up the code for button A and B presses, displaying the reaction time, and testing the game. Encourage students to challenge their peers and improve their reaction times.

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

- Microbit

Required equipment for this lesson:

- Microbit
- Traffic Lights Kit

- Phillips Screwdriver

Learning Goals	Learning Outcomes
1. Develop understanding of micro:bit and STOP:bit Traffic Lights for creating a reaction game. 2. Learn to add and utilise the "stopbit" extension in the MakeCode toolbox. 3. Understand and apply the concept of variables to measure reaction time. 4. Develop skills to program micro:bit buttons for specific actions. 5. Learn to display data on the micro:bit's LED matrix and interpret the results.	1. Develop a new project using MakeCode for micro:bit. 2. Add the 'stopbit' extension to the toolbox for programming the traffic lights kit. 3. Create and utilise variables 'startTime', 'endTime', and 'reactionTime' to measure reaction time. 4. Program button A to initiate the traffic light sequence and record the start time. 5. Program button B to record the end time and calculate the reaction time.

Lesson: Pedestrian Crossing

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare for this interactive lesson by familiarising yourself with the MakeCode editor for micro:bit and the STOP:bit traffic lights. Understand how to add the 'stopbit' extension and create a 'seconds' variable. Be ready to guide students through coding a traffic light sequence, including red light display, button press detection, and the full traffic light sequence. Ensure you know how to download the code to a micro:bit for testing.

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- Traffic Lights Kit
- Phillips Screwdriver

Learning Goals	Learning Outcomes
1. Understand and apply the use of MakeCode editor for micro:bit in creating a new project. 2. Learn to add and utilise the "stopbit" extension for programming the STOP:bit traffic lights kit. 3. Develop skills in creating and manipulating variables, specifically the 'seconds' variable in this context. 4. Gain proficiency in coding for specific outcomes such as displaying the red light and detecting button press on the micro:bit. 5. Master the sequence of traffic lights and their corresponding symbols on the micro:bit, and successfully download and test the code.	1. Develop a new project and attach STOP:bit traffic lights to the micro:bit. 2. Add the "stopbit" extension to the MakeCode editor toolbox. 3. Create a variable named 'seconds' for the countdown function. 4. Program the micro:bit to display a red light and an 'X' symbol at the start. 5. Implement button press detection to simulate a pedestrian waiting to cross. 6. Code a traffic light sequence with green, yellow, and red lights, along with appropriate wait times. 7. Download and test the code on the micro:bit.

Lesson: Build your Move Motor Sensor Car

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Ensure all materials are ready, including the Microbit and 4 AA batteries. Guide students through the step-by-step instructions provided in the yellow booklet, ensuring they understand each stage of assembly, connection to Makecode, and adding the Move Motor Extension. Facilitate their understanding of coding the motors, using the buzzer, Zip LEDs, line following sensors, and the distance sensor. Encourage exploration and experimentation once the Move Motor Sensor Car is built.

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- Move Motor Car

Learning Goals	Learning Outcomes
1. Develop practical skills in assembling a Move Motor Sensor Car. 2. Understand how to connect the Move Motor Sensor Car to Makecode. 3. Acquire coding skills for controlling the motors, buzzer, and LEDs of the Move Motor Sensor Car. 4. Learn to utilise the line following and distance sensors for navigation. 5. Encourage exploration and creativity in coding for different movements and LED usage.	1. Identify and organise components of the Move Motor Sensor Car kit. 2. Assemble the Move Motor Sensor Car following the provided instructions. 3. Connect the assembled car to Makecode and add the Move Motor Extension. 4. Code the motors, buzzer, Zip LEDs, line following sensors, and distance sensor of the car. 5. Apply learned skills to explore and create new movements and LED patterns.

Lesson: Line Following Car

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

In this lesson, students will program a Move Motor Car to follow a line track using a Microbit. They will create a new project on the MakeCode website, add the kitronik-move-motor extension, and create variables for the left and right line sensors and their difference. Students will then program the car to turn right, left, and move forward based on these sensor readings. After testing their code on a track, students can tweak the code to improve the car's speed and performance on more complex tracks.

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- Move Motor Car

Learning Goals	Learning Outcomes
1. Understand and apply the concept of programming a Microbit to control a Move Motor Car. 2. Create and manipulate variables to store sensor values and control the car's movements. 3. Implement conditional logic to guide the car's movements based on sensor readings. 4. Use LEDs for visual feedback and enhance the functionality of the car. 5. Experiment with code modifications to optimise the car's performance on different tracks.	1. Programme the Move Motor Car to follow a line track using a Microbit. 2. Create a new project on the https://makecode.microbit.org website. 3. Add the kitronik-move-motor extension to the project and utilise the custom blocks to program the Move Motor car. 4. Create and utilise variables to store values of the left and right line sensors and their difference. 5. Set up the LEDs on the Move Motor car to light up different colours depending on the car's direction. 6. Programme the car to turn right when the left sensor reads a higher darker value than the right sensor. 7. Programme the car to turn left when the right sensor reads a higher darker value than the left sensor. 8. Programme the car to move forwards when the left and right sensors have similar readings. 9. Test the programmed car on a track and observe its autonomous driving. 10. Tweak the code to improve the car's speed and performance on different tracks.

Lesson: Move Motor Measure

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

This lesson involves using a sonar sensor on a Move Motor car to measure distances and transmit the data to another Microbit. Students will learn how an ultrasonic sensor works and how to program two Microbits to communicate with each other. They will need to add specific extensions to their project, set units and radio groups, and create code to measure and display distances. The lesson requires hands-on work with Microbits and the Move Motor car, and also includes online coding using the makecode.microbit.org website.

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- Move Motor Car

Learning Goals	Learning Outcomes
1. Understand the function and operation of an ultrasonic sensor. 2. Develop proficiency in programming Microbits for specific tasks. 3. Gain skills in using radio groups for communication between Microbits. 4. Learn to measure distances using an ultrasonic sensor and a Microbit. 5. Acquire the ability to display measurements on a separate Microbit.	1. Understand and explain the function of an ultrasonic sensor and how it measures distance. 2. Program the Microbit inside the Move Motor car to measure distances and send the measurements to another Microbit. 3. Add necessary extensions to the project and set the units for measurement. 4. Program the second Microbit to send a "measure" message and display the received measurement. 5. Successfully test the functionality of the system by measuring and displaying distances.

Lesson: Traffic Lights and Car Communication

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

This lesson involves coding a set of traffic lights and a robot car using Microbits. Students will program the traffic lights to display a sequence and broadcast the light being shown. The robot car will receive this broadcast and decide whether to stop or go. The lesson involves creating two code projects, adding a 'stopbit' extension, programming a sequence, broadcasting the state, programming the car, receiving the message, downloading the code, and an additional challenge. Teachers should ensure they have the necessary equipment and familiarise themselves with the coding platforms used.

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- Traffic Lights Kit
- Move Motor Car

Learning Goals	Learning Outcomes
1. Understand and apply the concept of radio communication between Microbits. 2. Program a sequence of traffic light signals using code blocks. 3. Develop skills to broadcast and receive specific messages based on traffic light states. 4. Control the movement of a robot car based on received messages. 5. Enhance problem-solving skills by modifying the code to respond based on the proximity of the car to the traffic lights.	1. Code a set of traffic lights to run through a sequence and broadcast the displayed light. 2. Program a robot car to receive the broadcast and decide whether to stop or go based on the traffic light signal. 3. Use the "stopbit" extension to create custom code blocks for programming the traffic lights kit. 4. Program the car to move at different speeds or stop, depending on the received message from the traffic lights. 5. Modify the code to make the car respond to the traffic lights based on its proximity to them.

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Robotics & Engineering Short Course / Module: Intermediate Robotics & Engineering

Intermediate Robotics & Engineering

Lesson: Assemble the Smart Home

□ Beginner	□ 120 mins

Required equipment for this lesson:

- Smart Home Kit

Learning Goals	Learning Outcomes

Lesson: LED Lights

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- Smart Home Kit

Learning Goals	Learning Outcomes

Lesson: Temperature & Humidity Sensor

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- Smart Home Kit

Learning Goals	Learning Outcomes

Lesson: Motion Sensor

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- Smart Home Kit

Learning Goals	Learning Outcomes

Lesson: Moisture Sensor

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- Smart Home Kit

Learning Goals	Learning Outcomes

Lesson: Gas Sensor

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit

- Smart Home Kit

Learning Goals	Learning Outcomes

Lesson: Electronic Display (LCD)

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- Smart Home Kit

Learning Goals	Learning Outcomes

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Robotics & Engineering Short Course / Module: Advanced Robotics & Engineering

Advanced Robotics & Engineering

Lesson: Rain Detector

□ Advanced	□ 120 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- Smart Home Kit

Learning Goals	Learning Outcomes

Lesson: Smart Climate System

□ Advanced	□ 120 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- Smart Home Kit

Learning Goals	Learning Outcomes

Lesson: Solar Renewable Energy

□ Advanced	□ 120 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- Smart Home Kit

Learning Goals	Learning Outcomes

Lesson: Gas Leakage Detector

□ Advanced	□ 120 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- Smart Home Kit

Learning Goals	Learning Outcomes

Lesson: Home of the Future

□ Advanced	□ 180 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- Smart Home Kit

Learning Goals	Learning Outcomes

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Robotics & Engineering Short Course / Module: Advanced Robotics & Engineering

Advanced Robotics & Engineering

Lesson: Autonomous Car

□ Intermediate	□ 60 mins	□ Student Quiz	□ Student Challenge

Prepare to introduce students to the concept of ultrasonic sensors and how they function. Guide them through creating a new project on the MakeCode website, adding the kitronik-move-motor extension. Assist them in programming the sensor to measure distance and display it on the Microbit. Progress to programming the car to maintain a 10cm distance from an object, including reversing. Finally, challenge students to improve the code, adding lights and randomised movement to enhance the car's obstacle avoidance.

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- Move Motor Car

Learning Goals	Learning Outcomes
1. Understand the function and operation of ultrasonic sensors. 2. Develop skills in creating a new project using the kitronik-move-motor extension. 3. Acquire the ability to program a sensor to measure distance. 4. Learn to code a car to maintain a specific distance from an object. 5. Enhance problem-solving skills by programming the car to reverse and maintain distance.	1. Understand the function and operation of an ultrasonic sensor. 2. Create a new project on the MakeCode Microbit website and add the necessary extension. 3. Program the sensor to measure and display the distance to an object. 4. Modify the code to make the car maintain a distance of 10cm from an object. 5. Enhance the code to reverse the car until it is exactly 10cm away from an object. 6. Program the car to free roam and avoid objects by stopping, reversing, and turning right when an object is detected within 10cm. 7. Improve the code for better navigation and add lights for visual feedback.

Lesson: Tilt Remote Control Car

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

In this lesson, students will learn to control a Move Motor car using a Microbit as a remote controller. They will create two code projects: one for the remote control and another for the car. The lesson involves programming the Microbit to detect tilts in different directions and send corresponding messages to the car. The students will also add code to stop the car and to light up the LEDs on the car in different colours. Ensure each remote and car set uses a different radio group to avoid crossed signals.

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- Move Motor Car

Learning Goals	Learning Outcomes
1. Understand and apply the concept of radio communication between two Microbits. 2. Programme a Microbit to send specific messages based on different gestures. 3. Develop the ability to programme a Move Motor car to respond to different messages received. 4. Test and debug the code to ensure the car responds correctly to the remote control. 5. Extend the project by adding additional features such as LED light changes.	1. Programme a Microbit as a remote control to send directional commands. 2. Programme a Microbit to receive and execute directional commands in a Move Motor car. 3. Test and debug the code to ensure correct functioning of the remote-controlled car. 4. Download and implement the code onto the Microbits. 5. Extend the code to include LED light changes in response to different commands as an additional challenge.

Lesson: Attach the Move Motor Klaw

□ Advanced	□ 30 mins	□ Student Quiz

Ensure to guide students in seeking adult assistance for the complex parts of constructing the Move Motor Sensor Car. Facilitate the unpacking process, ensuring all necessary items are present. Lastly, guide students through the instruction booklet, assisting them in attaching the Move Motor Klaw to the car, either vertically or horizontally.

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- Move Motor Car
- Phillips Screwdriver
- Move Motor Klaw

Learning Goals	Learning Outcomes
1. Develop the ability to follow complex instructions with adult supervision. 2. Understand the process of unpacking and organising components for assembly. 3. Gain practical skills in using tools such as a small Phillips head screwdriver. 4. Learn to assemble the Move Motor Klaw and attach it to the Move Motor Car. 5. Understand the flexibility of design in attaching the Klaw either vertically or horizontally.	1. Identify and gather necessary components for building the Move Motor Klaw. 2. Correctly open the package and organise its contents. 3. Follow the provided instructions to assemble the Move Motor Klaw. 4. Successfully attach the Move Motor Klaw to the Move Motor Car. 5. Demonstrate safe and effective use of tools during assembly.

Lesson: Rescue Bot

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- Move Motor Car
- Move Motor Klaw

Learning Goals	Learning Outcomes

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Robotics & Engineering Short Course / Module: Advanced Robotics & Engineering

Advanced Robotics & Engineering

Lesson: Meet Your Raspberry Pi Pico

□ Advanced	□ 20 mins	□ Student Quiz

The lesson introduces the Raspberry Pi Pico W, a versatile microcontroller board ideal for learning electronics, programming, and IoT projects. Teachers should familiarize themselves with the Pico's components, including the RP2040 chip, Micro-USB port, GPIO pins, and ADC pins. The lesson also covers the RP2040 chip's features, such as its dual-core ARM Cortex-M0+ processor, RAM, flash memory, and communication protocols (UART, SPI, I2C).

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Raspberry Pi PICO W

Learning Goals	Learning Outcomes
• Understand the purpose and functionality of the Raspberry Pi Pico W microcontroller board. • Identify and describe the various components on the Raspberry Pi Pico W board. • Understand the features and functions of the RP2040 chip, including its processor, RAM, flash memory, and communication protocols. • Understand the role and usage of the GPIO pins on the Raspberry Pi Pico W. • Understand how to use the ADC pins to read analog signals. • Understand the function of the BOOTSEL button and how to use it to upload programs to the Raspberry Pi Pico W. • Learn how to use the built-in LED for testing and debugging programs.	• By the end of the lesson, students will be able to identify and describe the main components of the Raspberry Pi Pico W, including the RP2040 chip, the GPIO pins, the ADC pins, and the BOOTSEL button. • Students will understand the function and application of the RP2040 chip, including its processor, RAM, flash memory, and communication protocols (UART, SPI, I2C). • Students will be able to explain the purpose and use of the GPIO pins on the Raspberry Pi Pico W, including their role in interfacing with external components and special functions like PWM and UART. • Students will be able to describe the function of the ADC pins and how they can be used to read analog signals from various sensors. • Students will understand the function of the BOOTSEL button and how it is used to upload programs to the Raspberry Pi Pico W.

Lesson: Setting Up Your Pico

□ Advanced	□ 30 mins	□ Student Quiz

In this lesson, teachers will guide students through the process of setting up a Raspberry Pi Pico W. The lesson starts with an introduction to the Pico and its capabilities, followed by downloading the necessary firmware. Teachers will then demonstrate how to connect the Pico to a computer using a micro USB cable and how to transfer the firmware onto the Pico. The lesson will then move onto downloading and installing the Thonny editor, a software used to program the Pico using MicroPython language. Teachers will also guide students on how to select the correct Python version and install the necessary libraries. The lesson concludes with students having a fully set up and ready to program Pico.

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- USB Cable
- Raspberry Pi PICO W

Learning Goals	Learning Outcomes
• Understand the purpose and function of the Raspberry Pi Pico W and how to connect it to a computer. • Learn how to download and install the necessary firmware for the Raspberry Pi Pico W. • Master the process of connecting the Raspberry Pi Pico W to a computer using a micro USB cable and the BOOTSEL button. • Develop skills in using a file manager to transfer firmware files to the Raspberry Pi Pico W. • Gain knowledge on how to download, install, and use the Thonny editor for programming the Raspberry Pi Pico W using the MicroPython language. • Learn how to install necessary libraries, such as the picozero library, to aid in programming the Raspberry Pi Pico W.	• By the end of the lesson, students will be able to download and install the latest firmware for the Raspberry Pi Pico W. • Students will be able to correctly connect their Raspberry Pi Pico W to a computer using a micro USB cable. • Students will be able to successfully transfer the downloaded firmware onto their Raspberry Pi Pico W. • Students will be able to download, install and open the Thonny editor on their computer. • Students will be able to select the correct Python version (MicroPython) in the Thonny editor for programming their Raspberry Pi Pico W. • Students will be able to install the necessary picozero library in the Thonny editor for completing future projects.

Lesson: Pico's First Blink

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

This lesson is designed to introduce students to the Raspberry Pi Pico and its capabilities. Teachers should ensure that students have access to a Raspberry Pi Pico and a micro USB cable for this lesson. The lesson will guide students through setting up their Pico, using the Thonny editor to write and run a simple Python script, and understanding the role of libraries in coding. The main activity is creating a script to make the onboard LED blink. Teachers should encourage students to experiment with different blink speeds and discuss how the code can be modified to achieve this. The lesson concludes with safely disconnecting the Pico.

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- USB Cable
- Raspberry Pi PICO W

Learning Goals	Learning Outcomes
• Understand the purpose and process of connecting a Raspberry Pi Pico to a computer and setting up the Thonny editor. • Learn how to select the correct interpreter for the Raspberry Pi Pico in Thonny. • Develop the ability to create a new file in Thonny and understand the importance of naming conventions in coding. • Understand the concept of libraries in Python and how to import them into a project. • Learn how to write Python code to control the onboard LED of the Raspberry Pi Pico, including turning it on and making it blink. • Understand the concept of functions and loops in Python, and how to use them to control the blinking speed of the LED.	• Upon completion of the lesson, students will be able to connect a Raspberry Pi Pico to a computer and set up the Thonny editor for programming. • Students will be able to import necessary libraries for controlling the onboard LED on the Raspberry Pi Pico. • Students will be able to write and execute a Python script to turn the onboard LED on. • Students will be able to create a Python function to toggle the state of the LED. • Students will be able to use a loop to continuously call the function, thereby making the LED blink at a specified interval. • Students will be able to adjust the blink speed of the LED by changing the delay time in the code.

Lesson: Wifi Signal Strength Scanner

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

Make sure students have completed the previous lessons in the Pico Basics module. Students should be familiar with basic Python programming and have a working knowledge of Raspberry Pico.

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

- Chromebook/Laptop/PC

Learning Goals	Learning Outcomes
1. Understand the basics of Wi-Fi scanning and signal strength measurement. 2. Develop skills in programming a Wi-Fi signal strength scanner using Python. 3. Learn to create and serve web content displaying real-time Wi-Fi data. 4. Gain experience in setting up and managing a simple web server on a microcontroller. 5. Enhance problem-solving abilities through practical application and testing of the scanner.	1. Import required libraries for Wi-Fi scanning and web server functionality. 2. Configure Wi-Fi credentials correctly for network connection. 3. Implement a function to scan and retrieve nearby Wi-Fi networks' SSID and RSSI. 4. Generate an HTML webpage displaying Wi-Fi signal strength data in a table format. 5. Set up and run a web server to serve the Wi-Fi signal strength webpage.

Lesson: IoT Web-Controlled LED Light

□ Advanced	□ 60 mins	□ Student Quiz	□ Student Challenge

This lesson assumes students have already set up Thonny editor and are familiar with the Raspberry Pi Pico. Make sure students have access to a WiFi network and know the SSID and password.

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- USB Cable
- Raspberry Pi PICO W

Learning Goals	Learning Outcomes
1. Understand the integration of IoT devices with web technologies. 2. Develop skills in programming a Raspberry Pi Pico for network connectivity. 3. Learn to create and serve a web page that interacts with hardware. 4. Gain proficiency in using Python for socket programming and web server implementation. 5. Apply knowledge to control physical devices remotely via a web interface.	1. Import required libraries for IoT project. 2. Connect Raspberry Pi Pico to WiFi network. 3. Create and open a socket on the device. 4. Generate HTML for web page with LED control. 5. Serve web page and control LED based on user input.

Lesson: Controlling a Servo

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit

- Smart Home Kit

Learning Goals	Learning Outcomes

Lesson: Controlling a Motor

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- Smart Home Kit

Learning Goals	Learning Outcomes

Lesson: Using Solar Power

□ Beginner	□ 60 mins	□ Student Quiz	□ Student Challenge

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

- Chromebook/Laptop/PC

Required equipment for this lesson:

- Microbit
- Smart Home Kit

Learning Goals	Learning Outcomes

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Robotics & Engineering Short Course / Module: Classroom Based Assessments (CBAs)

Classroom Based Assessments (CBAs)

Lesson: Project Planning

□ Intermediate	□ 120 mins

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

- Chromebook/Laptop/PC

Learning Goals	Learning Outcomes

Lesson: Project Development

□ Intermediate	□ 240 mins

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

- Chromebook/Laptop/PC

Learning Goals	Learning Outcomes

Lesson: Project Presentation and Evaluation

□ Intermediate	□ 120 mins

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

- Chromebook/Laptop/PC

Learning Goals	Learning Outcomes

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Robotics & Engineering Short Course / Module: Classroom Based Assessments (CBAs)

Classroom Based Assessments (CBAs)

Teacher Learning Plan / Digital Skills Curriculum 2025/26 / Robotics & Engineering Short Course / Module: Classroom Based Assessments (CBAs)

Classroom Based Assessments (CBAs)