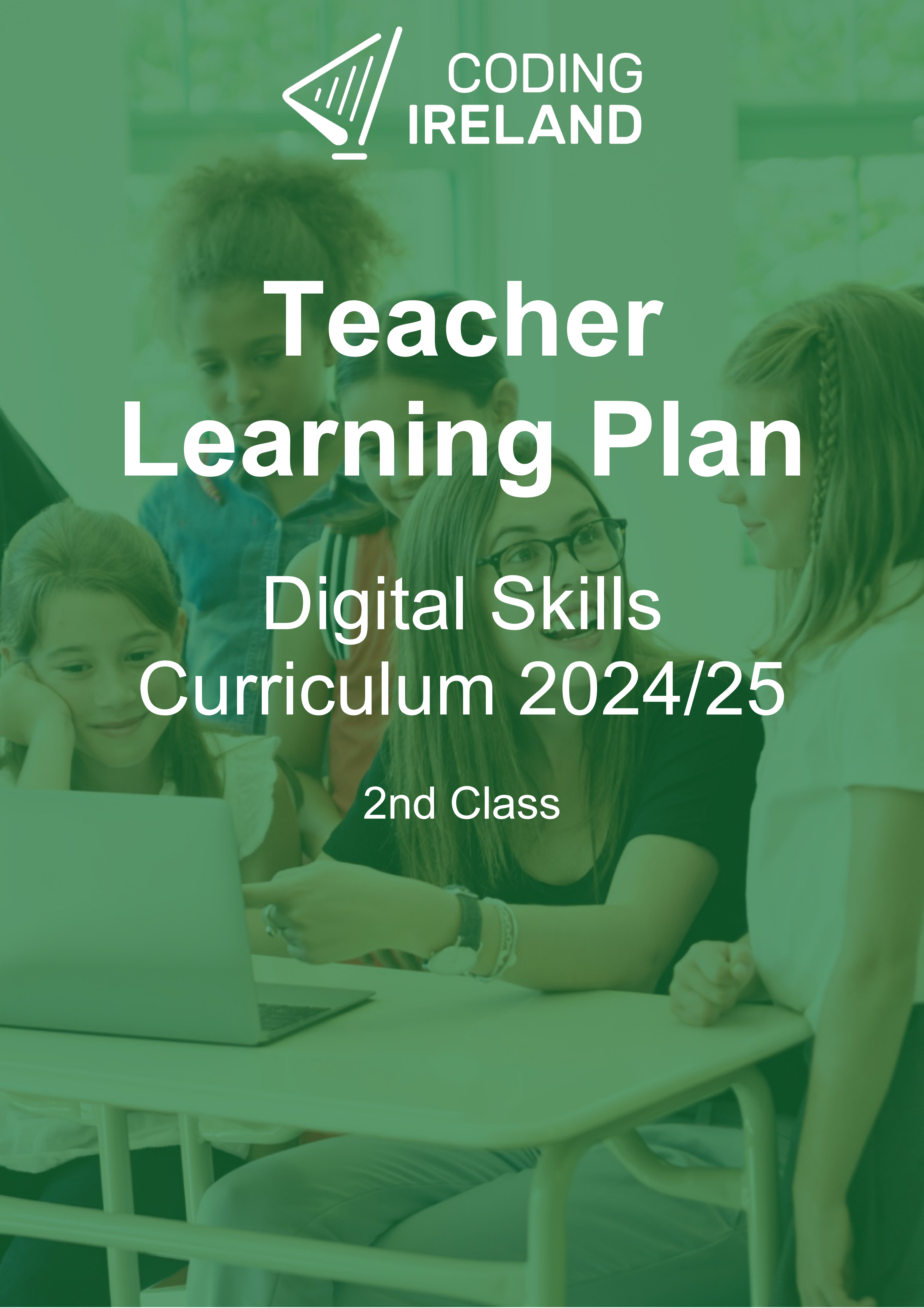
CODING IRELAND



Teacher Learning Plan

Digital Skills Curriculum 2024/25

2nd Class

Table of Contents

How to Use This Learning Plan

This learning plan provides an overview of all the modules available for 2nd Class, 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.

Module: My First Coding Adventure

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

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

Week 1

Lesson: Introduction to Coding

□ Beginner	□ 20 mins	System.Threading.Tasks.Task`1[System.String]

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

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

Lesson: Scratch Tutorial

□ Beginner	□ 40 mins	System.Threading.Tasks.Task`1[System.String]	□ Student Quiz

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

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

- Chromebook/Laptop/PC
- iPad/Tablet

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

Week 2

Lesson: A Dancing Cat and Bat

□ Beginner	□ 30 mins	System.Threading.Tasks.Task`1[System.String]	□ Student Quiz	□ Student Challenge

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

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

- Chromebook/Laptop/PC
- iPad/Tablet

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

Module: Exploring Coding

This module introduces students to coding through Scratch, a block-based visual programming language. The module is divided into eight modules, each focusing on a different aspect of Scratch. The lessons range from creating interactive projects with sound effects to designing games and stories. Teachers should ensure students understand each step before proceeding, encourage creativity and experimentation, and facilitate group brainstorming sessions to foster teamwork. The module concludes with students creating their own Scratch projects, reinforcing the skills they've learned throughout The module.

Duration	Equipment
6 weeks	Students can use any of these devices: • Chromebook/Laptop/PC • iPad/Tablet
Module Goals	**Module Outcomes**
1. Master the use of Scratch's sound blocks to animate sprites with unique sounds. 2. Develop skills in creating simple games using Scratch, including sprite animation and mouse cursor interaction. 3. Understand and apply Scratch's pen blocks, colour changing, and mouse movement features to create a drawing tool. 4. Learn to control sprite movement using keyboard keys, detect sprite collisions, and manipulate sprite visibility in Scratch. 5. Gain proficiency in creating and manipulating clones of a sprite, making them appear at random positions, and incorporating sound effects in Scratch.	1. Develop a Scratch project with animal sprites that produce unique sounds when clicked. 2. Create a game in Scratch where a sprite follows the mouse cursor. 3. Design a rainbow paintbrush in Scratch using pen blocks, colour changing, and mouse movement features. 4. Build a game in Scratch where a shark sprite catches fish sprites, using keyboard controls and sprite collision detection. 5. Construct an interactive 'Balloon Pop' game in Scratch, utilising sprite clones, random positioning, and click-triggered sound effects. 6. Create a simple story in Scratch, incorporating backdrop changes, character dialogue, and teleportation. 7. Develop a simple game in Scratch, manipulating characters, controlling movement with keyboard keys, and enabling character interaction. 8. Brainstorm, receive feedback, and initiate a group project in Scratch, fostering creativity and teamwork.

Week 1

Lesson: Animal Sounds

□ Beginner	□ 40 mins	System.Threading.Tasks.Task`1[System.String]	□ Student Quiz	□ Student Challenge

In this lesson, students will learn to use Scratch's sound blocks to animate sprites with unique sounds. They will create a new Scratch project, add a backdrop, and then add a series of animal sprites. For each sprite, they will code a unique sound to play when the sprite is clicked. The lesson will guide students through adding a cat, dog, parrot, and horse sprite, each with their own sound. The lesson concludes with students testing their project by clicking on each animal to hear its sound. This lesson provides a fun, interactive way to explore coding with Scratch.

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 animal sounds and how they can be represented in a coding project. • Learn how to create a new project on Scratch and add sprites to the project. • Develop the ability to add backdrops to a Scratch project to enhance its visual appeal. • Gain skills in coding sounds to sprites, enabling them to play unique sounds when clicked. • Experience the process of testing and debugging a Scratch project to ensure it functions as intended. • Enhance creativity and problem-solving skills by exploring different ways to use sounds and sprites in Scratch.	• By the end of the lesson, students will be able to create a new Scratch project and add a backdrop to it. • Students will be able to add different animal sprites to their Scratch project. • Students will be able to assign unique sounds to each animal sprite in their project. • Students will be able to write code in Scratch to make the animal sprites play their sounds when clicked. • Students will be able to test their project by clicking on each animal sprite and listening to the sounds they make. • Students will be able to understand the concept of interactivity in Scratch projects, as demonstrated by the sounds played when the sprites are clicked.

Week 2

Lesson: Crabby Cursor Chaser

□ Beginner	□ 40 mins	System.Threading.Tasks.Task`1[System.String]	□ Student Quiz	□ Student Challenge

In this lesson, students will learn how to create a simple game using Scratch. They will learn how to add and animate a sprite, and make it follow the mouse cursor. The lesson starts with an introduction to the game and its objective. It then guides students through creating a new Scratch project, adding a crab sprite, and coding it to point towards and follow the mouse cursor. Students will also learn to animate the crab by switching between its costumes. Finally, they will add a backdrop to set the scene for their game. The lesson concludes with a wrap-up encouraging students to continue practicing and experimenting with coding.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• Students will learn how to create a new project in Scratch and add a new sprite to it. • Students will develop the ability to animate a sprite and make it follow the mouse cursor. • Students will understand how to use the 'forever' block to continuously execute a set of instructions. • Students will gain knowledge on how to use the 'point towards' and 'move' blocks to control the movement of a sprite. • Students will learn how to switch between different costumes of a sprite to create an animation effect. • Students will learn how to add a backdrop to their Scratch project to enhance the visual appeal of their game.	• By the end of the lesson, students will be able to create a new project on Scratch and add a new sprite to it. • Students will be able to write code in Scratch to make the sprite point towards the mouse cursor. • Students will be able to animate the sprite to make it move towards the mouse cursor. • Students will be able to use different costumes to animate the sprite, making it look like it's opening and closing its claws. • Students will be able to add a backdrop to their Scratch project to set the scene for their game. • Students will be able to understand the concept of continuous movement in coding through the use of the 'forever' block in Scratch.

Week 3

Lesson: Rainbow Paintbrush

□ Beginner	□ 40 mins	System.Threading.Tasks.Task`1[System.String]	□ Student Quiz	□ Student Challenge

In this lesson, students will learn how to create a rainbow paintbrush using Scratch's pen blocks, color changing, and mouse movement features. They will start by creating a new project and adding a ball sprite. They will then program the sprite to follow the mouse pointer and add the Pen extension to enable drawing. Students will learn to use the 'pen down' block to draw lines and the 'erase all' block to clear the drawing. Finally, they will add the 'change pen color by (10)' block to create a rainbow effect. The lesson concludes with a wrap-up and encouragement to continue exploring Scratch.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• Understand and apply the use of Scratch's pen blocks to create a drawing tool. • Learn how to manipulate sprites to follow mouse movements on the Scratch platform. • Develop skills in adding and utilizing extensions in Scratch, specifically the Pen extension. • Apply coding concepts to create interactive features, such as clearing the drawing with the space bar. • Understand and implement the use of colour changing blocks to create a rainbow effect in drawings. • Develop an attitude of exploration and creativity in using coding to create interactive tools and features.	• By the end of the lesson, students will be able to create a new project in Scratch, demonstrating their understanding of the Scratch interface. • Students will be able to add and manipulate a sprite in Scratch, showcasing their ability to interact with the sprite library and apply code to a sprite. • Students will be able to use the mouse-pointer feature in Scratch, demonstrating their ability to control sprite movement with the mouse. • Students will be able to implement the Pen extension in Scratch, indicating their understanding of how to add and use extensions in Scratch. • Students will be able to use the pen down, erase all, and change pen color by blocks in Scratch, demonstrating their ability to control drawing, erasing, and color changing functions in Scratch. • By the end of the lesson, students will have created a rainbow paintbrush tool in Scratch, showcasing their ability to combine different Scratch features and blocks to create a

Week 4

Lesson: Shark Tank

□ Intermediate	□ 40 mins	System.Threading.Tasks.Task`1[System.String]	□ Student Quiz	□ Student Challenge

The 'Shark Tank' lesson is a fun, interactive way for students to learn coding basics using Scratch. Teachers should familiarize themselves with the Scratch platform and the specific blocks of code used in this lesson. The lesson involves creating a game where a shark, controlled by the arrow keys, tries to catch a fish. When the shark catches the fish, the fish disappears. This lesson introduces the 'touching' block to detect sprite collisions. Teachers should ensure students understand each step before moving on, and encourage experimentation and creativity. The final product is a simple, playable game that students can continue to modify and improve.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• Students will understand how to create a new project in Scratch and add sprites and backdrops to their project. • Students will learn how to control the movement of sprites using the arrow keys on their keyboard. • Students will be able to use the touching block to detect when one sprite collides with another. • Students will learn how to make a sprite disappear when it is caught by another sprite. • Students will gain experience in troubleshooting and debugging their code. • Students will develop an understanding of the iterative process of coding and the importance of practice in improving their skills.	• By the end of the lesson, students will be able to create a new project in Scratch. • Students will be able to add and manipulate sprites in a Scratch project, including changing their size and position. • Students will be able to add a backdrop to a Scratch project to create a specific setting. • Students will be able to program sprite movement in Scratch, including using the arrow keys for control. • Students will be able to use the 'touching' block in Scratch to detect when one sprite collides with another. • Students will be able to create a simple game in Scratch where a sprite disappears when caught by another sprite.

Week 5

Lesson: Balloon Pop Game

□ Intermediate	□ 40 mins	System.Threading.Tasks.Task`1[System.String]	□ Student Quiz	□ Student Challenge

These teaching notes will guide you through facilitating the 'Balloon Pop Game' lesson. Begin by introducing the game and its objectives to the students. Then, guide them through creating a new Scratch project and adding the balloon sprite. Explain the concept of clones and how to create them in Scratch. Next, show students how to make the balloon clones appear at random positions and move around the screen. Finally, teach them how to add a popping effect and sound when the balloons are clicked. Wrap up the lesson by congratulating the students on their achievement and encouraging them to continue practicing their Scratch skills.

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 creating and controlling clones in Scratch programming. • Learn how to make sprites appear at random positions on the screen. • Develop skills to make sprites move and bounce around the screen. • Understand how to add interactivity to a game by making sprites respond to mouse clicks. • Learn how to add sound effects to enhance the gaming experience. • Develop creativity and problem-solving skills by designing and coding a simple game.	• By the end of this lesson, students will be able to create a new project on Scratch and add a specific sprite to it. • Students will be able to use the 'create clone of myself' block to create multiple copies of a sprite. • Students will be able to use the 'when I start as a clone' block to control the behaviour of cloned sprites. • Students will be able to use the 'go to random position' block to make sprites appear at different locations on the screen. • Students will be able to use the 'glide' and 'if on edge, bounce' blocks to make sprites move around the screen. • Students will be able to use the 'when this sprite clicked', 'start sound', and 'delete this clone' blocks to create an interactive game element where sprites disappear and play a sound when clicked.

Week 6

Lesson: Story Adventures

□ Intermediate	□ 40 mins	System.Threading.Tasks.Task`1[System.String]	□ Student Quiz	□ Student Challenge

In this 'Story Adventures' lesson, students will create a simple story using Scratch. They will learn to change backdrops, and make them talk. The lesson will guide students through creating a new project, adding backdrops and characters, setting the start scene, and coding character dialogue and movements. Teachers should ensure students understand each step before moving on to the next. Encourage creativity and experimentation with different backdrops, sprites, and storylines. Remind students that practice improves coding skills. Celebrate their completion of the lesson and the creation of their unique story.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• Understand and apply the basics of creating a story using Scratch, including changing backdrops, controlling a character's movements, and making the character speak. • Develop the ability to create a new project in Scratch and navigate its interface. • Learn to use the 'say' block from the Looks toolbox to make a character speak in the story. • Understand the use of the 'switch backdrop to' block from the Looks toolbox to change the scene in the story. • Learn to use the 'wait' block from the Control toolbox to control the timing of events in the story. • Develop creativity and storytelling skills by creating a unique story adventure with different backdrops, sprites, and story lines.	• By the end of the lesson, students will be able to create a new project on Scratch. • Students will be able to add and change backdrops in a Scratch project. • Students will be able to add a sprite to their Scratch project. • Students will be able to use the 'when green flag clicked' and 'switch backdrop to' blocks to set the start scene of their story. • Students will be able to use the 'say' block to make their sprite speak. • Students will be able to use the 'wait' and 'switch backdrop to' blocks to make their sprite teleport to a different scene.

Week 7

Lesson: Dino food game

□ Intermediate	□ 40 mins	System.Threading.Tasks.Task`1[System.String]	□ Student Quiz	□ Student Challenge

In this lesson, students will create a simple game using Scratch, where they control a dinosaur to catch beetles. Teachers should familiarize themselves with the Scratch platform and the specific steps of the lesson. The lesson involves adding characters, controlling their size and movement, and adding a backdrop.

The game also includes interaction between characters and random positioning of sprites. Teachers should make sure students understand the purpose of each block of code they add and encourage them to experiment with different settings and sprites. The lesson concludes with a wrap-up discussion about the skills learned and their potential applications.

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

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
• By the end of this lesson, students will be able to create and manipulate characters in Scratch. • Students will understand how to control character movement using keyboard inputs in Scratch. • Students will learn how to use conditional statements to create interactions between characters in Scratch. • Students will be able to use random functions to create unpredictability in game play. • Students will learn how to add and manipulate backdrops in Scratch to enhance the visual appeal of their game. • Students will develop an understanding of the iterative process of game design, including testing and refining their game.	• By the end of the lesson, students will be able to create a new project on Scratch, demonstrating their ability to navigate the platform. • Students will be able to add and manipulate characters in Scratch, including changing their size and adding movement commands. • Students will be able to use the arrow keys to control character movement in their Scratch game, demonstrating understanding of basic coding commands. • Students will be able to program character interactions in Scratch, such as having one character 'eat' another when they touch. • Students will be able to add a backdrop to their Scratch game, demonstrating their ability to customize the game environment. • Students will be able to create a simple game in Scratch, demonstrating their understanding of the lesson's concepts and their ability to apply them in a practical project.

Week 8

Lesson: Brainstorming Blast

□ Advanced	□ 60 mins	System.Threading.Tasks.Task`1[System.String]

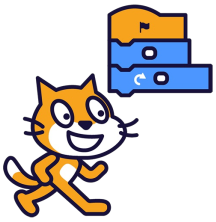
In this lesson, 'Brainstorming Blast', students will brainstorm ideas for their own Scratch projects. Start by introducing the lesson and demonstrating a simple Scratch project. Divide students into small groups for brainstorming, reminding them of the importance of teamwork. Set a timer for the brainstorming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

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

- Chromebook/Laptop/PC
- iPad/Tablet

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

Module: Creative Coding Challenges



This module provides a comprehensive guide to creative coding, focusing on animation and character interaction. Teachers should encourage students to experiment with different characters and backgrounds, and to test their animations regularly. The module gradually builds up complexity, culminating in a final project where students create their own animated story. Teachers should emphasise the importance of planning and refining their projects for the best results.

Duration	Equipment
6 weeks	Required Equipment: • Chromebook/Laptop/PC • iPad/Tablet
Module Goals	**Module Outcomes**
1. Master the art of animating characters and controlling their movements. 2. Develop proficiency in creating loops for repeated movements and actions. 3. Effectively add dialogue to characters and manage timing for smooth conversation flow. 4. Learn to animate backgrounds, create scene transitions, and manage character visibility. 5. Gain expertise in broadcasting signals to trigger actions and create interactive animations.	1. Animate characters by changing costumes and controlling their dance speed. 2. Code character movements, including walking and bouncing, and implement loops for repeated actions. 3. Create dynamic dialogues between characters, controlling timing for smooth conversation flow. 4. Design and animate a conversation between characters, including speech recording and mouth animation. 5. Animate moving backgrounds and create smooth scene transitions. 6. Develop a hide-and-seek game with characters that appear and disappear. 7. Master the use of broadcasts to trigger events and interactions in an animated story. 8. Create an original animated story, applying learned skills for character animation, dialogue, scene control, and transitions.

Week 1

Lesson: Animating Characters: Moving and Changing Costumes

□ Beginner	□ 40 mins	System.Threading.Tasks.Task`1[System.String]	□ Student Quiz	□ Student Challenge

In this step-by-step lesson, you'll animate a character, making them dance by changing their costume every second. You'll add a character, set a stage background, use multiple costumes for animation, control the dance speed, and test your animation. You'll also have the opportunity to add more characters for a bigger dance party.

Required equipment for this lesson:

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand how to add characters and backgrounds to a project. 2. Learn to change character costumes to create animation. 3. Master the use of loops to repeat animations continuously. 4. Control the speed of costume changes to create rhythmic dance moves. 5. Develop skills to animate multiple characters simultaneously.	1. Select and add a character from the sprite library to the project. 2. Choose and apply a stage background from the library. 3. Utilise the 'Costumes' tab to change character costumes and create animation. 4. Implement a 'forever' loop to continuously repeat the dance animation. 5. Control the speed of costume changes to create rhythmic dance moves. 6. Test the animation by observing the character's dance on the stage. 7. Add and animate additional characters to enhance the dance party.

Week 2

Lesson: Loop it! Making Movements Repeat

□ Beginner	□ 40 mins	System.Threading.Tasks.Task`1[System.String]	□ Student Quiz	□ Student Challenge

In this step-by-step lesson, you'll animate the Scratch Cat to walk across the screen. You'll create a new project, code the cat's movements, change its costumes for a walking effect, control the speed, and loop the walk. You'll also learn how to make the cat bounce at the edge and animate a race for an extra challenge.

Required equipment for this lesson:

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand and apply the concept of loops in animation. 2. Create and control character movement using Scratch programming. 3. Manipulate character costumes to simulate walking or running. 4. Implement control structures to manage animation speed and direction. 5. Apply learned skills to create a competitive animation scenario.	1. Create a new Scratch project and initialise a sprite. 2. Code the sprite to move across the screen using the "move 10 steps" block. 3. Implement costume changes to simulate walking animation. 4. Control the speed of the animation using the "wait 0.2 seconds" block. 5. Apply the "forever" block to loop the walking animation continuously. 6. Use the "if on edge, bounce" block to make the sprite turn around at the screen edge. 7. Animate a race between two different sprites, adjusting their speed and movement.

Week 3

Lesson: Talking and Timing: Adding Dialogue and Delays

□ Beginner	□ 40 mins	System.Threading.Tasks.Task`1[System.String]	□ Student Quiz	□ Student Challenge

In this step-by-step lesson, you'll learn how to create a dynamic conversation between two characters using Scratch. You'll add dialogue to your sprites, use timing to ensure each character speaks at the right moment, and test your project to ensure a smooth conversation flow.

Required equipment for this lesson:

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Develop skills to add dialogue to sprites in a Scratch project. 2. Understand the use of timing to control the sequence of dialogue between sprites. 3. Learn to create a new project and add characters in Scratch. 4. Gain knowledge on setting a suitable scene for the dialogue. 5. Master the process of testing and adjusting the conversation for smooth flow.	1. Develop a Scratch project featuring two characters engaging in a dialogue. 2. Utilise the sprite library to select and add characters to the project. 3. Choose an appropriate background to set the scene for the conversation. 4. Apply the "say" and "wait" blocks to create a timed conversation between the characters. 5. Test and adjust the conversation's flow by modifying the timing in the "wait" blocks.

Week 4

Lesson: Animating a Conversation: Kiran and Ripley in Space

□ Beginner	□ 40 mins	System.Threading.Tasks.Task`1[System.String]	□ Student Quiz	□ Student Challenge

Encourage students to work collaboratively and take turns recording dialogue for Kiran and Ripley. Familiarise yourself with Scratch’s sprite library and costume-switching features beforehand. Emphasise the importance of timing mouth movements with recorded speech. Be ready to assist with troubleshooting, such as adjusting "wait" blocks or refining animations for smoother conversations.

Required equipment for this lesson:

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Master the use of Scratch platform for creating animations. 2. Develop skills in planning and scripting a conversation for animation. 3. Acquire the ability to animate characters in sync with recorded dialogue. 4. Learn to control the timing of animations to match the dialogue. 5. Enhance collaborative skills through refining and improving the animation with a partner.	1. Create a new Scratch project and set a space-themed background. 2. Add the characters Kiran and Ripley from the sprite library. 3. Plan and record a conversation between the characters Kiran and Ripley. 4. Animate the characters' mouths to synchronise with the recorded dialogue. 5. Test, refine, and improve the conversation animation in collaboration with a partner.

Week 5

Lesson: Backgrounds in Motion: Scene Transitions

□ Beginner	□ 40 mins	System.Threading.Tasks.Task`1[System.String]	□ Student Quiz	□ Student Challenge

Encourage students to visualise the concept of dynamic scene transitions by relating it to real-world examples, like a bus journey. Familiarise yourself with Scratch’s sprite and backdrop libraries beforehand. Demonstrate each step clearly, ensuring students understand the purpose of each block. Allow time for testing and troubleshooting, fostering creativity and experimentation.

Required equipment for this lesson:

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand how to animate a moving background in Scratch projects. 2. Learn to create smooth scene transitions between different backdrops. 3. Develop skills in coding for sprite movement and positioning. 4. Gain knowledge on how to use control blocks for repeating actions. 5. Acquire the ability to test and adjust animation settings for optimal visual effect.	1. Manipulate Scratch City Bus sprite and set its starting position in a chosen background. 2. Animate the bus to glide smoothly across the screen. 3. Implement a transition to a new background, creating an illusion of the bus moving to a new location. 4. Utilise the "repeat" block to enable the bus to drive through multiple backgrounds. 5. Optionally animate the background itself for a more dynamic effect and test the animation for smooth transitions.

Week 6

Lesson: Hide and Seek: Making Characters Appear and Disappear

□ Beginner	□ 40 mins	System.Threading.Tasks.Task`1[System.String]	□ Student Quiz	□ Student Challenge

Encourage students to explore Scratch features confidently. Familiarise yourself with the "hide," "show," and "go to random position" blocks, as these are key. Test the game yourself to ensure smooth functionality. Highlight creativity by suggesting unique characters and backgrounds. Remind students to save their work regularly and troubleshoot timing or positioning issues collaboratively.

Required equipment for this lesson:

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand and apply the concept of interactive character movement in a digital game. 2. Develop skills in coding character actions such as hiding and reappearing at random positions. 3. Enhance creativity by designing a unique hide-and-seek game environment. 4. Learn to incorporate sound effects to enhance user engagement in a game. 5. Extend the complexity of the game by adding multiple characters.	1. Understand and apply the concept of making characters disappear and reappear in a game setting. 2. Effectively add and code multiple characters in an interactive game. 3. Set an appropriate background for the game scenario. 4. Apply timing controls to manage character appearances. 5. Enhance game experience by adding sound effects and additional characters (optional).

Week 7

Lesson: Broadcasting Signals: Triggering Actions Like a Pro

□ Beginner	□ 40 mins	System.Threading.Tasks.Task`1[System.String]	□ Student Quiz	□ Student Challenge

Dive into the enchanting world of fairy tales while mastering the use of broadcasts in Scratch projects. Learn to trigger events for a seamless story flow, create a fairy tale scene, add characters, and code them to interact perfectly. Enhance your story with sound effects, background changes, and more, all controlled by broadcasts. By the end, you'll be adept at creating interactive and engaging animations.

Required equipment for this lesson:

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Master the use of broadcasting to control the timing of events in Scratch projects. 2. Develop skills in setting up a scene and characters in a Scratch project. 3. Understand how to use broadcasts to trigger character interactions and story progression. 4. Learn to add more complexity to the story by using broadcasts to trigger additional details like sound effects or background changes. 5. Enhance creativity and storytelling skills by creating a fairy tale scene with interactive characters.	1. Utilise broadcasting to control the timing of events in Scratch projects. 2. Apply the 'hide' and 'show' functions to manage character appearances in a story. 3. Use 'broadcast' and 'when I receive' blocks to trigger character actions and dialogue. 4. Integrate sound effects and backdrop changes to enhance storytelling. 5. Create an interactive fairy tale scene using a combination of broadcasts, character actions, and dialogue.

Week 8

Lesson: The Big Finale: Create Your Own Animated Story!

□ Beginner	□ 60 mins	System.Threading.Tasks.Task`1[System.String]

Encourage students to reflect on the skills they’ve learned and how to apply them creatively. Prepare examples of animated stories to inspire ideas and provide guidance during story selection. Reinforce key animation techniques like dialogue, scene transitions, and broadcasts. Allow time for testing and refining, and celebrate students’ completed projects.

Required equipment for this lesson:

- Chromebook/Laptop/PC
- iPad/Tablet

Learning Goals	Learning Outcomes
1. Develop an original animated story or adapt a classic fairy tale using the animation skills learned. 2. Utilise character animation techniques, including movement, costume changes, and dialogue. 3. Effectively use broadcasts for scene control and character interaction. 4. Apply techniques for character appearance and disappearance to enhance the story. 5. Create smooth scene transitions to progress the narrative.	1. Select and conceptualise a story for animation, either an existing fairy tale or an original narrative. 2. Apply animation skills to characters, including movement, costume changes, and dialogue. 3. Utilise broadcasts for scene control and to trigger key events in the narrative. 4. Implement techniques for characters to appear and disappear, and for smooth scene transitions. 5. Test, refine, and finalise the animated story, ensuring proper timing, dialogue flow, and animation aesthetics.

Module: Discovering the Magic of Circuits

This module involves guiding students through the fascinating world of circuits using Snap Circuits. Each week, you'll introduce new concepts, from understanding electricity and switches to exploring motors, LED lights, and phototransistors. The hands-on nature of The module requires you to ensure students understand the function and placement of each component. Encourage exploration and experimentation after each lesson to reinforce learning. Safety is paramount, so ensure all components are handled correctly.

Duration	Equipment
3 weeks	Required Equipment: • Snap Circuits
Module Goals	**Module Outcomes**
1. Understand and apply the principles of electricity and switches using Snap Circuits. 2. Develop practical skills in assembling and operating circuits with different components such as a battery holder, wire block, lamp block, and switch block. 3. Gain knowledge about motors and LED lights, and their application in circuit construction. 4. Learn about phototransistors and their role in light-controlled circuits. 5. Explore the concept of electrical conduction and distinguish between conductors and insulators.	1. Understand and demonstrate the function of a switch in an electrical circuit using Snap Circuits. 2. Construct and operate an electric light circuit controlled by a press switch. 3. Assemble a circuit with a motor fan and press switch, demonstrating the operation of a motor in a circuit. 4. Build a basic LED circuit, understanding the importance of the direction of electricity flow. 5. Construct a light-responsive circuit using a phototransistor and LED light block, demonstrating the role of light in controlling electricity. 6. Identify and classify objects as conductors or insulators through experimentation with a self-assembled circuit.

Week 1

Lesson: Electric Light and Switch

□ Beginner	□ 30 mins	System.Threading.Tasks.Task`1[System.String]

Prepare for this interactive lesson by familiarising yourself with the Snap Circuits tool. You'll guide students through understanding electricity, switches, and how they work together to light up a bulb. The lesson involves hands-on activities like assembling a circuit on a base grid, inserting batteries, and placing components like a battery holder, wire block, lamp block, and switch block. Ensure students understand the grid system for placing components. The lesson culminates in lighting up the assembled circuit, providing a practical demonstration of the concepts learned.

Required equipment for this lesson:

- Snap Circuits

Learning Goals	Learning Outcomes
1. Understand the basic principles of electricity and its everyday applications. 2. Identify and explain the function of a switch in an electrical circuit. 3. Construct a simple electrical circuit using Snap Circuits, demonstrating practical skills and safety awareness. 4. Apply knowledge of electricity and circuits to explain why a light bulb lights up when connected to a battery and controlled by a switch. 5. Develop problem-solving skills by troubleshooting and correcting any issues with the circuit to ensure the light bulb lights up.	1. Understand and explain the function of an electric light and switch. 2. Assemble a Snap Circuit correctly following the provided steps. 3. Identify and correctly place the components of a Snap Circuit: base grid, battery holder, wire block, lamp block, and switch block. 4. Demonstrate the operation of a Snap Circuit by successfully lighting up the lamp. 5. Apply the concept of electricity flow in a circuit to a real-world analogy.

Week 2

Lesson: Electric Light and Press Switch

□ Beginner	□ 30 mins	System.Threading.Tasks.Task`1[System.String]

For this lesson, you'll be guiding students through the process of building an electric light circuit with a press switch. Begin with a refresher on electricity and switches, then move onto the practical application using Snap Circuits. Ensure students understand the grid system for placement of components. As they build the circuit, explain the function of each part. Once the circuit is complete, they should be able to light up the lamp using the press switch. Encourage exploration and experimentation after the main task, reinforcing learning through play.

Required equipment for this lesson:

- Snap Circuits

Learning Goals	Learning Outcomes
1. Understand the basic principles of electricity and its flow. 2. Recognise the function and importance of a switch in controlling the flow of electricity. 3. Construct a simple circuit using a press switch and a light bulb. 4. Apply knowledge of circuits to successfully light up a bulb using a press switch. 5. Explore and experiment with different circuit configurations using Snap Circuits.	1. Understand and explain the function of a press switch in an electrical circuit. 2. Correctly assemble a basic circuit using Snap Circuits, including a battery holder, wire block, lamp block, and press switch block. 3. Demonstrate the operation of the assembled circuit by successfully lighting up the lamp using the press switch. 4. Identify and correct any issues in the circuit assembly that prevent the lamp from lighting up. 5. Apply the knowledge gained to explore and create different circuit configurations using the Snap Circuits set.

Week 3

Lesson: Motor Fan and Press Switch

□ Beginner	□ 30 mins	System.Threading.Tasks.Task`1[System.String]

This lesson involves guiding students through the process of building a circuit using a Snap Circuits set, ultimately powering a motor to make a fan fly. Start by explaining the concepts of electricity and motors, then guide students through each step of the circuit construction, from preparing the base grid to inserting batteries, placing the battery holder, wire block, motor block, and press switch block. Ensure students understand the function of each component. Attach the fan and demonstrate how the press switch activates the motor. Conclude with an exploration period, encouraging students to experiment with different circuits. Safety warnings should be emphasised throughout.

Required equipment for this lesson:

- Snap Circuits

Learning Goals	Learning Outcomes
1. Understand the concept of electricity and how it powers a motor. 2. Learn to assemble a simple circuit using a base grid, battery holder, wire block, motor block, and press switch block. 3. Develop skills in following step-by-step instructions to build a functioning circuit. 4. Appreciate the importance of safety when working with electrical circuits and moving parts. 5. Encourage exploration and application of learned concepts to create different circuits.	1. Understand the concept of electricity and how it powers a motor. 2. Identify and correctly place the base grid, battery holder, wire block, motor block, and press switch block on the Snap Circuits set. 3. Demonstrate safe handling of the Snap Circuits set, particularly the motor block and fan. 4. Successfully construct a circuit that powers a motor to make a fan fly. 5. Apply knowledge of circuits to explore and create new configurations with the Snap Circuits set.

Week 4

Lesson: LED Circuit

□ Beginner	□ 30 mins	System.Threading.Tasks.Task`1[System.String]

This lesson involves building a basic LED circuit and understanding the direction of electricity flow. Teachers should guide students through each step, from setting up the base grid to inserting batteries, placing the battery holder, wire block, LED light block and press switch block. Emphasise the importance of correct placement, particularly for the LED light block. Encourage students to experiment with LED direction and explore different circuit configurations. Reinforce learning by engaging with students during their exploration.

Required equipment for this lesson:

- Snap Circuits

Learning Goals	Learning Outcomes
1. Understand the basic principles of an LED circuit, including the direction of electricity flow. 2. Develop practical skills in assembling a basic LED circuit using Snap Circuits. 3. Recognise the importance of correct placement of circuit components, particularly the LED light block. 4. Apply knowledge of circuitry to experiment with and troubleshoot LED circuits. 5. Develop a curiosity for further exploration and understanding of electrical circuits.	1. Construct a basic LED circuit using Snap Circuits. 2. Identify and correctly place the components of an LED circuit including the battery holder, wire block, LED light block, and press switch block. 3. Demonstrate understanding of the direction of electricity flow in a circuit by correctly placing the positive and negative sides of the LED light block. 4. Explain the effect of reversing the direction of the LED light block on the functioning of the circuit. 5. Apply knowledge of circuits to experiment with and build different configurations using Snap Circuits.

Week 5

Lesson: LED Circuit and Phototransistor

□ Beginner	□ 30 mins	System.Threading.Tasks.Task`1[System.String]

Prepare to introduce phototransistors and their function in a circuit. Ensure a stable workspace for each group and verify the presence of batteries in each Snap Circuits set. Guide students through placing the battery holder, wire block, red LED light block, and phototransistor block on the base grid, emphasising correct placement for circuit functionality. Demonstrate the circuit in action by shining a light into the phototransistor block. Encourage exploration and reinforce concepts learned throughout the lesson.

Required equipment for this lesson:

- Snap Circuits

Learning Goals	Learning Outcomes
1. Understand the function and application of phototransistors in circuits. 2. Develop skills in assembling a circuit using a base grid, battery holder, wire block, LED light block, and phototransistor block. 3. Recognise the importance of correct placement and direction of circuit components for proper electricity flow. 4. Observe and explain how light can control electricity in a circuit. 5. Apply knowledge and skills to explore and create different circuits.	1. Understand the function and application of a phototransistor in a circuit. 2. Correctly assemble a circuit using a base grid, battery holder, wire block, LED light block, and phototransistor block. 3. Identify and correctly position the positive and negative sides of the LED light block and phototransistor block in a circuit. 4. Demonstrate how light can complete a circuit and activate an LED light by shining a light into the phototransistor block. 5. Apply knowledge of circuits to explore and create different configurations using available blocks.

Week 6

Lesson: Conduction

□ Beginner	□ 30 mins	System.Threading.Tasks.Task`1[System.String]

In this lesson, students will explore the concept of electrical conduction through a hands-on experiment. They will build a circuit using a base grid, batteries, a battery holder, a red LED light block, a 100 Ω resistor block, and wire blocks. They will then test various objects and materials to determine if they conduct electricity. The lesson concludes with a review of their findings and a discussion about conductors and insulators. Ensure all components are handled safely and stored properly post-lesson.

Required equipment for this lesson:

- Snap Circuits

Learning Goals	Learning Outcomes
1. Understand the concept of electrical conduction and identify materials that can conduct electricity. 2. Build a basic circuit using a base grid, batteries, a battery holder, an LED light block, a resistor block, and wire blocks. 3. Recognise the importance of correct placement and connection of circuit components for successful electrical flow. 4. Conduct experiments to test the conductivity of various objects and materials. 5. Reflect on the results of the experiments, reinforcing the understanding of conductors and insulators.	1. Understand and explain the concept of electrical conduction. 2. Identify and correctly assemble components of a circuit including a battery holder, LED light block, resistor block, and wire blocks. 3. Conduct an experiment to test the conductivity of various objects and materials. 4. Classify objects and materials as conductors or insulators based on experimental results. 5. Disassemble and store circuit components correctly for future use.