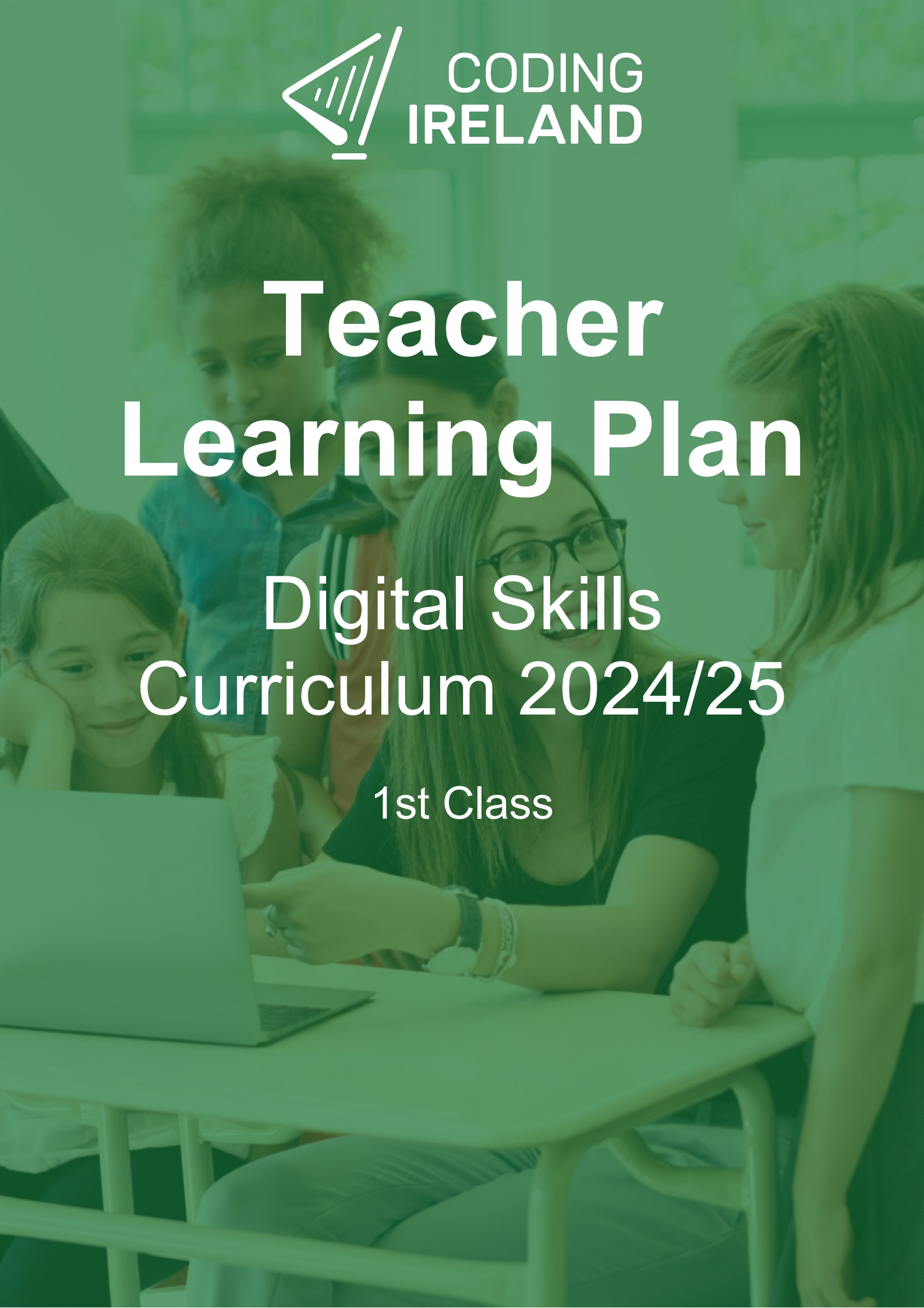


CODING IRELAND

Teacher Learning Plan

Digital Skills Curriculum 2024/25

1st Class

Table of Contents

How to Use This Learning Plan

This learning plan provides an overview of all the modules available for 1st 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.

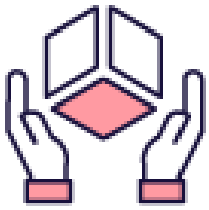
Student Access

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: The World of Logic



The World of Logic module is designed to introduce students to essential logical thinking skills through engaging and interactive activities. Starting with fundamental pattern recognition, students will advance to understanding categorization and identifying outliers. Teachers will guide students through sequencing tasks, reinforcing the importance of order and direction with fun grid games and maze challenges. The module concludes with an introduction to algorithms, where students apply their learning by creating simple dance routines. This module aims to build foundational critical thinking skills, essential for problem-solving in computer science and coding.

Duration	**Equipment**
3 weeks	Required Equipment: • Interactive Display • Pen & Paper • Printer
Module Goals	**Module Outcomes**
1. Develop skills in identifying and completing patterns. 2. Enhance understanding of categorisation and outlier identification. 3. Improve ability to sequence events and follow instructions. 4. Master the concept of mapping and navigating grids. 5. Understand and apply the principles of algorithms in creating sequences.	1. Identify and complete patterns using games and worksheets. 2. Understand and apply the concept of categorisation and outlier identification through interactive tasks. 3. Demonstrate understanding of sequencing and following instructions through games and worksheets. 4. Apply logical thinking and problem-solving skills in navigating and mapping grids. 5. Create and understand algorithms through the creation of dance routines.

Week 1

Lesson: Pattern Play: Spot the Sequence

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

Prepare to introduce the game 'Simon Says' to students, emphasising the importance of listening for the phrase 'Simon says'. Transition into interactive whiteboard games focusing on pattern recognition with themes of fruit, animals, numbers, and letters. Distribute a 'Complete the Pattern' worksheet for independent work, encouraging students to fill in missing parts of sequences. Conclude by reviewing patterns and discussing students' thought processes in completing them.

Required equipment for this lesson:

- Printer
- Interactive Display

Learning Goals	Learning Outcomes
1. Develop attentive listening skills through the 'Simon Says' game. 2. Identify and understand patterns and sequences in different contexts such as fruits and animals. 3. Apply knowledge of numerical and alphabetical order to complete sequences. 4. Enhance problem-solving skills by completing missing parts of patterns on a worksheet. 5. Articulate reasoning behind pattern completion choices during class discussions.	1. Identify and follow instructions in the 'Simon Says' game. 2. Recognise and predict patterns in the 'Data Detectives (Fruit)' game. 3. Apply pattern recognition skills to different themes in the 'Data Detectives (Animals)' game. 4. Order numbers correctly in the 'Data Detectives (Numbers)' game. 5. Sequence letters accurately in the 'Data Detectives (Letters)' game. 6. Complete missing parts of sequences in the 'Complete the Pattern' worksheet. 7. Explain reasoning behind pattern completion choices during the 'Conclusion' discussion.

Week 2

Lesson: Which One's Different? The Odd One Out

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

Prepare for an engaging lesson on categorisation and identification of outliers. Start with a fun game of 'Word Tennis' to warm up the class. Then, use a slideshow to display groups of items, asking students to identify the odd one out. Distribute worksheets for students to circle the odd item in each group and create their own exercise. Conclude by reinforcing the importance of categorisation in computer science and coding.

Required equipment for this lesson:

- Printer
- Interactive Display

Learning Goals	Learning Outcomes
1. Develop understanding of categorisation and identification of outliers. 2. Enhance critical thinking skills through interactive activities. 3. Identify items that do not belong in a group using visual aids. 4. Apply knowledge to complete a worksheet, identifying the odd one out in various groups. 5. Design their own 'Odd One Out' exercise, demonstrating understanding of the concept.	1. Identify and categorise items based on their characteristics during the 'Word Tennis' game. 2. Explain the reasoning behind the identification of outliers in the 'What doesn't Belong' slideshow. 3. Accurately identify the odd one out in each group on the 'What doesn't belong' worksheet. 4. Create a unique 'Odd One Out' exercise on the worksheet, demonstrating understanding of the concept. 5. Apply the concept of categorisation and outlier identification, fundamental to computer science and coding.

Week 3

Lesson: What Happens Next? First/Then Fun!

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

In this lesson, you'll guide students through a series of activities designed to enhance their understanding of sequencing and following instructions. Starting with a game of Simon Says, you'll then move onto sequencing daily events, followed by interactive games. The lesson concludes with a 'First/Then' worksheet, encouraging students to consider what comes next in a sequence of events.

Required equipment for this lesson:

- Printer
- Interactive Display

Learning Goals	Learning Outcomes
1. Develop skills in following and understanding instructions through interactive games. 2. Understand and apply the concept of sequencing in daily activities. 3. Enhance critical thinking skills by predicting subsequent events in a given scenario. 4. Improve communication skills by explaining reasoning behind chosen sequences. 5. Recognise the importance of sequencing and clear instructions in the context of coding.	1. Demonstrate understanding of following instructions through participation in a game of Simon Says. 2. Arrange pictures of daily events in the correct sequence to show understanding of sequencing. 3. Apply sequencing skills to arrange steps involved in daily activities such as putting on shoes, eating cereal, planting a seed, and reading a book. 4. Complete the 'First/Then' worksheet by identifying and illustrating what comes next after a given activity. 5. Explain the chosen sequence of events and justify the reasoning behind the choices made on the 'First/Then' worksheet.

Week 4

Lesson: Grid Games: Mapping the 3x3

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

Prepare for an engaging lesson that combines games and worksheets to teach students about following and giving instructions. Start with a game of Simon Says, emphasising the importance of listening to commands. Introduce the grid game, guiding characters to specific locations. Progress to an advanced version of the game, then move on to a 'Giving Directions' worksheet, where students guide a bird to its fruit. Conclude by summarising the activities and their relevance to logical thinking, problem-solving, and coding.

Required equipment for this lesson:

- Printer
- Interactive Display

Learning Goals	Learning Outcomes
1. Develop ability to follow and give precise instructions. 2. Understand the concept of grid mapping and its application. 3. Enhance logical thinking and problem-solving skills through interactive games. 4. Apply learned skills in a practical context through a worksheet activity. 5. Recognise the importance of precise instructions in the digital world and its relevance to programming.	1. Follow instructions accurately in the game of Simon Says. 2. Guide characters to specific locations on a grid using clear, step-by-step instructions in the Grid Game. 3. Progress to the advanced Grid Game, demonstrating increased proficiency in giving directions. 4. Complete the 'Giving Directions' worksheet by guiding a bird to its fruit using a sequence of instructions. 5. Reflect on the importance of precise instructions and its relevance to logical thinking, problem-solving, and coding.

Week 5

Lesson: Mastering the Maze: Navigating the 4x4

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

Embark on a journey to understand the importance of sequence and direction in coding. You'll guide a bird to its fruit on a 5x5 grid, take turns giving directions, and work independently on a worksheet. Remember, careful planning of your sequence is key to success. Let's get started!

Required equipment for this lesson:

- Printer
- Interactive Display

Learning Goals	Learning Outcomes
1. Develop an understanding of sequence and direction in the context of coding. 2. Apply navigational skills in a practical, hands-on activity. 3. Enhance problem-solving abilities through grid navigation tasks. 4. Work independently to plan and execute a sequence of directions. 5. Reflect on learning experiences and share strategies for planning and problem-solving.	1. Understand and apply the concepts of sequence and direction in a 5x5 grid navigation task. 2. Effectively communicate and follow directions as both a navigator and a robot in a classroom setting. 3. Successfully guide a bird to its fruit on a 5x5 grid using a sequence of directional commands. 4. Independently complete a 4x4 grid worksheet, demonstrating careful planning and double-checking of directions. 5. Reflect on learning, identify challenges, and share strategies for planning the bird's path in a group discussion.

Week 6

Lesson: Dance Commands: Creating Move Algorithms

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

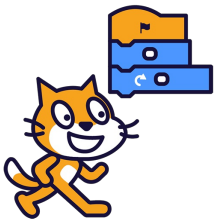
In this lesson, you'll guide students through understanding algorithms via interactive activities. Begin by explaining algorithms using a slideshow, then transition into creating "dance algorithms". Students will suggest dance moves to form a class dance routine. Afterwards, they'll work in small groups to create their own dance routines. The lesson concludes with a reflection on the importance of algorithms in coding.

Required equipment for this lesson:

- Pen & Paper
- Interactive Display

Learning Goals	Learning Outcomes
1. Understand the concept of algorithms as a set of instructions followed in a specific order. 2. Develop the ability to create and interpret simple algorithms using dance moves. 3. Enhance teamwork and collaboration skills through creating group dance algorithms. 4. Improve creativity by inventing unique dance move sequences. 5. Apply understanding of algorithms to real-world activities, laying a foundation for future coding lessons.	1. Understand and explain the concept of an algorithm as a set of instructions. 2. Create and demonstrate a dance routine by following a sequence of steps, representing an algorithm. 3. Develop and use symbols to represent different dance moves in an algorithm. 4. Work collaboratively in small groups to create a simple dance routine, demonstrating understanding of sequence in an algorithm. 5. Reflect on the concept of algorithms and its application in coding and everyday tasks.

Module: Adventures in Digital Creation



This module introduces students to the world of digital creation through Scratch Junior. Starting with basic motion blocks, students will progressively learn to create projects, change backgrounds, add characters, and control character speed. Each lesson concludes with a challenge and free play exploration time. Teachers should encourage creativity and experimentation, and be prepared to recap on previous lessons as necessary.

Duration	**Equipment**
4 weeks	Required Equipment: • iPad/Tablet
Module Goals	**Module Outcomes**
1. Master the basics of Scratch Junior, including project creation, interface exploration, and the use of basic motion blocks. 2. Develop skills in changing backgrounds and creating unique character routines within Scratch Junior. 3. Learn to add and control multiple characters within a project, including synchronising their movements. 4. Gain proficiency in controlling the size and movement of objects across different backgrounds. 5. Understand and apply advanced programming concepts such as controlling character speed, sequencing commands, and looping actions.	1. Manipulate basic motion blocks to move characters in Scratch Junior. 2. Create and modify backgrounds, and choreograph a dance routine for characters in Scratch Junior. 3. Add a second character in Scratch Junior and coordinate their movements. 4. Programme a car to drive across a city background and adjust its size in Scratch Junior. 5. Control character speed, sequence commands, and create a race with different sprites in Scratch Junior. 6. Programme a spaceman character to float in space using repeat actions in Scratch Junior.

Week 1

Lesson: Move and Groove: The Dance Party Begins!

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

In this lesson, teachers will guide students through the basics of coding using Scratch Jr. Students will learn to control character movements and create their own dance routines. They'll explore changing backgrounds, creating sequences, and understanding the concept of cause and effect. Teachers will need to demonstrate how to use the green flag block and movement blocks, and guide students in setting the stage for their dance party. The lesson concludes with a review and encouragement for students to apply their new skills in other areas.

Required equipment for this lesson:

- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand and utilise Scratch Jr. as a tool for coding and creative expression. 2. Grasp the concept of coding sequences using the green flag and movement blocks. 3. Develop skills in controlling character actions and movements through coding. 4. Apply creativity in setting the stage with different backgrounds for their projects. 5. Create a unique dance routine by combining different movement blocks.	1. Understand and utilise Scratch Jr. to create basic coding sequences. 2. Apply the use of basic movement blocks to control character actions. 3. Initiate coding sequences using the green flag block in Scratch Jr. 4. Manipulate backgrounds in Scratch Jr. to set the scene for their projects. 5. Create a dance routine for a character using a sequence of movement blocks.

Week 2

Lesson: City Cruise: Driving Through the Streets

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

Prepare action cards for a game of charades, explaining the rules to students. Introduce the car driving activity in Scratch Jr, explaining how they'll program a car to navigate city streets. Show how to remove Scratch the Cat from the project, then demonstrate adding a car character and changing the background to a city scene. Teach students how to create a sequence of movement blocks for the car. For advanced students, introduce resizing the car. Encourage exploration and creativity during free play, prompting with questions about other vehicles and their placement.

Required equipment for this lesson:

- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand and apply the process of removing and adding characters in a digital project. 2. Develop the ability to change backgrounds in a digital environment. 3. Gain skills in recreating a digital scenario with minimal guidance. 4. Acquire advanced skills in resizing characters within a digital project. 5. Enhance creativity and exploration by adding diverse elements to a digital scenario.	1. Demonstrate ability to remove characters from a project in the application. 2. Create a sequence involving a car character driving across a city background. 3. Recreate a given video sequence independently. 4. Apply advanced skills to resize the car character within the sequence. 5. Explore the application further by adding different types of vehicles and experimenting with their placement and movement.

Week 3

Lesson: Floating in Space: Looping with the Spaceman

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

This lesson guides students through a space adventure using Scratch Jr. They'll learn to programme a spaceman to float in space, introducing the concept of repeating actions with the repeat block. The lesson starts with a discussion on space movement, followed by practical steps in Scratch Jr. to create the floating spaceman. Students will then add a repeat block for continuous movement, customise their astronaut, and finally, create a 'space dance' using different sequences and loops.

Required equipment for this lesson:

- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand and apply the concept of loops in coding using the repeat block in Scratch Jr. 2. Program a spaceman character to simulate floating in space using a sequence of movement blocks. 3. Explore the concept of space movement and how it differs from movement on Earth. 4. Customise a character in Scratch Jr. using the camera feature to personalise the spaceman. 5. Apply creativity and coding skills to create a unique 'space dance' using different sequences of movements and loops.	1. Understand and apply the concept of repeating actions using the repeat block in Scratch Jr. 2. Programme a spaceman character to simulate floating in space using a series of movement blocks. 3. Utilise the repeat block to create an endless floating motion, demonstrating an understanding of loops in coding. 4. Create a "space dance" by experimenting with different sequences of movements and using the repeat block to create patterns or loops. 5. Customise the astronaut character in Scratch Jr., demonstrating creativity and personalisation in coding.

Week 4

Lesson: Dribble and Dunk: Basketball Fun with Scratch Jr.

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

Initiate the lesson by discussing how a ball moves, focusing on bouncing. Guide students to set the scene in Scratch Jr., selecting a basketball court background and adding a basketball sprite and Scratch the Cat. Demonstrate how to program the basketball to bounce and move using parallel coding. Students then program Scratch the Cat to mimic the basketball's movement. For advanced learners, introduce the challenge of programming the ball to shoot into the basket. Conclude by encouraging further experimentation and sharing of projects.

Required equipment for this lesson:

- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand and articulate the concept of how a ball moves, particularly in the context of basketball. 2. Select and utilise appropriate sprites and backgrounds in Scratch Jr. to create a basketball court scene. 3. Apply coding principles to program a basketball sprite to bounce and move across the court. 4. Develop parallel coding skills to synchronise the movement of two sprites, simulating the action of dribbling a basketball. 5. Extend coding skills to create a sequence that simulates shooting a basketball into a hoop, demonstrating problem-solving and creativity.	1. Understand and discuss the movement of a basketball when thrown or dribbled. 2. Select and add appropriate background and characters in Scratch Jr. for a basketball game scenario. 3. Program a basketball to bounce and move forward using parallel coding in Scratch Jr. 4. Program Scratch the Cat to mimic the movement of the basketball, creating a dribbling effect. 5. Extend the program to simulate shooting the basketball into the basket, for those seeking an additional challenge.

Week 5

Lesson: Ready, Set, Go! The Great Frog vs. Pig Race

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

Engage students in a discussion about animal movement, leading to the concept of programming a race between two animals with different speeds. Set the scene with a suitable background in Scratch Jr. Introduce the characters, a fast frog and a slow pig. Teach parallel coding, allowing the frog to jump and move forward simultaneously. Introduce the variable speed block to control the pig's speed. Start the race and observe the effects of different speeds and movements. For early finishers, add a third character to the race for an extra challenge.

Required equipment for this lesson:

- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand and discuss the different ways animals move and how this can be represented in a digital race. 2. Select and apply appropriate backgrounds in Scratch Jr. to enhance the realism of the race scenario. 3. Identify and utilise the unique movement characteristics of different animals in the programming of a digital race. 4. Develop and apply parallel coding skills to enable simultaneous actions in digital characters. 5. Use variable speed blocks in Scratch Jr. to control the speed of digital characters, reflecting their natural movement speeds.	1. Identify and discuss the different ways animals move and apply this knowledge to a programming context. 2. Select and set an appropriate background for a race scenario in Scratch Jr. 3. Choose and describe the characteristics of two different race characters, a frog and a pig, in Scratch Jr. 4. Apply the concept of parallel coding to make the frog character jump and move forward simultaneously. 5. Use the variable speed block to control the speed of the pig character, demonstrating an understanding of the concept of variable speeds in programming. 6. Observe and analyse the outcome of the race, reflecting on how different speeds and movements affect the result.

Week 6

Lesson: Talk It Out: Programming a Fun Conversation

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

Prepare to introduce students to Scratch Jr's animal characters and their sounds. Guide them through recording sounds for their chosen animal. Introduce a complex task of creating a classroom setting with a teacher-student interaction. Demonstrate recording dialogue and using the 'wait' function for natural conversation flow. Allow time for students to experiment with their scenes, encouraging creativity and reinforcement of learned skills.

Required equipment for this lesson:

- iPad/Tablet

Learning Goals	Learning Outcomes
1. Identify and replicate various animal sounds. 2. Record and attach sounds to characters in Scratch Jr. 3. Create a classroom scene with multiple characters in Scratch Jr. 4. Record and sequence dialogue between characters. 5. Apply the 'wait' function to control timing of dialogue. 6. Experiment with different scenarios and character interactions in Scratch Jr.	1. Identify and replicate various animal sounds using Scratch Jr. 2. Record and attach sounds to chosen characters in Scratch Jr. 3. Create a classroom scene with two characters, a teacher and a student, in Scratch Jr. 4. Record a dialogue between the teacher and student characters, including a question and response. 5. Apply the 'wait' function in Scratch Jr. to control the timing of the dialogue for a natural conversation flow.

Week 7

Lesson: Maze Master: Help Your Character Escape!

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

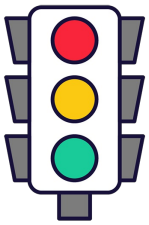
Prepare to engage students with the concept of mazes, guiding them to create their own in Scratch Jr. Demonstrate how to select and edit the background, draw a simple maze, and place a character. Show them how to use movement blocks to navigate the maze and encourage testing and adjusting their code. Extend the challenge by prompting them to create more complex mazes. Conclude by inviting students to share their mazes, highlighting the skills they've developed in creative design, problem-solving, and sequencing.

Required equipment for this lesson:

- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand and apply the concept of mazes in Scratch Jr. 2. Develop skills in editing and designing backgrounds in Scratch Jr. for maze creation. 3. Master the use of movement blocks to guide a character through a maze. 4. Enhance problem-solving abilities by creating and navigating through complex mazes. 5. Share and discuss their mazes, reflecting on challenges encountered and solutions developed.	1. Create and edit a maze background in Scratch Jr. 2. Position a character at the starting point of the maze. 3. Use movement blocks to navigate the character through the maze. 4. Design and navigate through a more complex maze. 5. Present and explain the created maze to the class.

Module: Electricity and Simple Circuits



This module involves guiding students through the understanding and application of electricity and simple circuits using Snap Circuits. Teachers will facilitate hands-on activities, including assembling circuits and experimenting with various components such as switches, LED lights, motors, and phototransistors. Emphasise safety, correct component placement, and the function of each part. Encourage exploration and reinforce learning through play. Familiarise yourself with Snap Circuits and prepare for each lesson by ensuring all necessary components are available.

Duration	**Equipment**
3 weeks	Required Equipment: • Snap Circuits
Module Goals	**Module Outcomes**
1. Understand the basic principles of electricity and its components such as switches, LED lights, motors, and phototransistors. 2. Assemble and disassemble simple circuits using Snap Circuits, demonstrating practical application of theoretical knowledge. 3. Identify and explain the function of different components within a circuit, including switches, LED lights, motors, and phototransistors. 4. Experiment with different circuit configurations and understand the impact of component placement and direction of electricity flow. 5. Explore the concepts of electrical conduction, identifying conductors and insulators through hands-on experimentation.	1. Assemble and operate a simple circuit using a switch block and lamp block. 2. Construct and control an electric light circuit using a press switch. 3. Build and activate a motor fan circuit using a press switch. 4. Create a basic LED circuit, understanding the importance of electricity flow direction. 5. Construct a light-responsive circuit using a phototransistor and LED light block. 6. Assemble a circuit and identify electrical conductors and insulators through experimentation.

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.