CODING IRELAND

Teacher Learning Plan

Digital Skills Curriculum 2024/25

Senior Infants

Table of Contents

How to Use This Learning Plan

This learning plan provides an overview of all the modules available for Senior Infants, 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

Senior Infants students do not log into the platform. All lessons are teacher-led, with the teacher delivering the content and guiding students through activities.

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: Solving Problems

This module focuses on problem-solving skills, primarily through sequencing and conditional logic. Engage students with interactive games like 'Simon Says' and 'Robot Roger', and reinforce learning with worksheets. Use the whiteboard for visual demonstrations and remember to vary themes to maintain interest. Encourage students to learn at their own pace and ensure they understand each concept before moving on.

Duration	Equipment
4 weeks	Required Equipment: • Interactive Display • IWB/Projector/Large Screen • Printer
Module Goals	**Module Outcomes**
1. Understand and apply the concept of sequencing through 'first/then' statements. 2. Comprehend and utilise 'if/then' statements to understand cause and effect. 3. Master the skill of putting events in order through various activities and games. 4. Learn to navigate and follow instructions using numbers and colours in a step-by-step manner. 5. Develop problem-solving skills and spatial awareness through grid navigation activities.	1. Apply 'first/then' statements and 'Bossy Words' to understand sequencing in coding. 2. Utilise 'if/then' statements to comprehend conditional logic through interactive activities. 3. Demonstrate understanding of event sequencing through games and worksheet activities. 4. Follow and give instructions for navigation using colours and numbers in games and worksheet activities. 5. Navigate a character on a 4x4 grid using up and down arrows, focusing on numbers, fruits, and colours. 6. Enhance problem-solving skills and spatial awareness by navigating a character on a 5x5 grid with various themes.

Week 1

Lesson: Bossy Words

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

In this lesson, you'll explore the concept of sequencing through 'first/then' statements, a fundamental skill in coding. You'll learn about 'Bossy Words', play 'Simon Says', discuss instructions, and complete a 'First/Then' worksheet. You'll also engage in activities involving slideshows and a 'First/Then Cut and Stick Activity' to reinforce your understanding.

Required equipment for this lesson:

- Printer
- Interactive Display

Learning Goals	Learning Outcomes
1. Develop understanding of 'Bossy Words' as instructions or commands. 2. Enhance ability to identify and use 'Bossy Words' in context. 3. Understand the importance of following instructions in daily life. 4. Grasp the concept of sequencing through 'first/then' statements. 5. Apply 'first/then' sequencing in practical activities.	1. Identify and understand the concept of 'Bossy Words' as instructions or commands. 2. Participate in the 'Simon Says' game, demonstrating comprehension of instructions. 3. Discuss and provide examples of instructions followed in daily life. 4. Understand and apply the concept of 'first/then' in sequencing everyday activities. 5. Complete the 'First/Then Cut and Stick Activity', demonstrating ability to sequence events logically.

Lesson: First Things First: What Comes Next? Version 2

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

In this alternative version of the lesson, teachers will guide students through activities to understand the concept of sequencing everyday activities using 'first/then' statements. The lesson begins with a discussion on 'Bossy Words' without slides, a game of 'Simon Says' with added sequencing elements, a discussion about instructions, and a 'First/Then' worksheet activity where students match and create sequences. Teachers should explain concepts in a simple, clear, and engaging manner, making learning fun and relatable to everyday life.

Required equipment for this lesson:

- IWB/Projector/Large Screen

Learning Goals	Learning Outcomes

Week 2

Lesson: Cause and Effect: If This Happens, Then What?

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

In this lesson, you'll guide your students through interactive activities to understand 'if/then' statements. You'll start with an 'If/Then' Simon Says game, followed by an interactive whiteboard game. Finally, students will complete an 'If/Then' activity worksheet to reinforce their understanding of conditional logic.

Required equipment for this lesson:

- Printer
- Interactive Display

Learning Goals	Learning Outcomes
1. Understand and apply the concept of 'if/then' statements through interactive activities. 2. Develop critical thinking skills by predicting outcomes in various 'if' scenarios. 3. Enhance communication skills by discussing and explaining their choices during activities. 4. Improve creativity and expression through drawing responses in the 'If/Then' worksheet activity. 5. Build foundational knowledge of conditional logic, a key concept in coding.	1. Understand and apply the concept of 'if/then' statements through interactive activities. 2. Participate effectively in an 'If/Then' version of Simon Says, demonstrating comprehension of conditional instructions. 3. Engage with an 'If/Then' interactive whiteboard game, making appropriate choices based on given scenarios. 4. Complete an 'If/Then' activity worksheet, demonstrating the ability to predict outcomes based on given conditions. 5. Exhibit understanding of conditional logic, a fundamental concept in coding, through discussion and explanation of their choices.

Week 3

Lesson: Sequence Master: Putting Events in Order

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

This lesson guides you, the teacher, in introducing the concept of sequencing to your students. You'll start with a 'Follow the Leader' game, move on to whiteboard sequencing activities, and use interactive games. The lesson concludes with a sequencing worksheet activity for students to apply their learning.

Required equipment for this lesson:

- Printer
- Interactive Display

Learning Goals	Learning Outcomes
1. Understand and apply the concept of sequencing through interactive activities and games. 2. Develop the ability to follow instructions and mimic actions in a 'Follow the Leader' game. 3. Engage in critical thinking by explaining reasoning when completing sequences. 4. Apply sequencing skills in a practical context by arranging events in the correct order on a worksheet. 5. Express creativity and reinforce learning by creating a personal sequence of events.	1. Imitate actions accurately in the 'Follow the Leader' game. 2. Understand and explain the concept of sequencing. 3. Complete sequencing activities correctly using interactive games. 4. Apply sequencing skills to arrange images in the correct order on a worksheet. 5. Create a personal sequence of events, demonstrating understanding and creativity.

Week 4

Lesson: Number Path: Step-by-Step Navigation

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

In this interactive lesson, you'll start by playing 'Robot Roger', a game that teaches you to follow and give instructions. You'll then move on to direction games, using colours and numbers to guide a frog to a target. Finally, you'll complete a worksheet activity, moving a counter on a number line based on given instructions.

Required equipment for this lesson:

- Printer
- Interactive Display

Learning Goals	Learning Outcomes
1. Understand and apply the concept of forward and backward movement on a number line. 2. Develop skills in giving and following instructions accurately. 3. Enhance problem-solving abilities and logical thinking through interactive games. 4. Improve spatial awareness and sequencing skills. 5. Apply knowledge of addition and subtraction in a practical context.	1. Understand and apply the concept of forward and backward movement on a number line. 2. Follow and give clear, specific instructions in the context of a game. 3. Develop spatial awareness and sequencing skills through direction games. 4. Apply basic coding concepts in a practical, interactive setting. 5. Use a number line to solve problems involving addition and subtraction.

Week 5

Lesson: Grid Navigator: Cracking the 4x4 Code

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

This lesson begins with a fun game of 'Simon Says' to practice following instructions. You'll then explore a 4x4 grid on an interactive whiteboard, learning how to navigate a character using up and down arrows. You'll play different versions of a grid game, focusing on numbers, fruits, and colours. Finally, you'll complete a worksheet activity involving a frog navigating a 4x4 grid to find fruit.

Required equipment for this lesson:

- Printer

Learning Goals	Learning Outcomes
1. Develop ability to follow instructions through interactive games. 2. Understand and apply the concept of a 4x4 grid. 3. Utilise directional language to navigate a character on a grid. 4. Identify and differentiate between numbers, fruits, and colours in a grid format. 5. Apply grid navigation skills to solve worksheet problems independently.	1. Follow instructions accurately during a game of Simon Says. 2. Understand and navigate a 4x4 grid using up and down arrows. 3. Apply grid navigation skills to different themed interactive games. 4. Identify and name numbers, fruits, and colours in the context of a 4x4 grid game. 5. Complete a grid worksheet independently, using directional instructions to locate items.

Week 6

Lesson: Grid Masters: Exploring the 5x5 Adventure

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

In this interactive lesson, you'll navigate a character on a 5x5 grid, enhancing your problem-solving skills and spatial awareness. You'll play a fun 'Robot Statues' game, review the 4x4 grid, and then move onto the 5x5 grid. You'll explore different themes like letters, animals, and sports, learning at your own pace.

Learning Goals	Learning Outcomes
1. Develop critical thinking and problem-solving skills through navigating a character on a 5x5 grid. 2. Enhance understanding of directions and spatial awareness through interactive games and activities. 3. Improve command recognition and response through participation in the 'Robot Statues' game. 4. Reinforce knowledge of 4x4 grid navigation as a foundation for understanding 5x5 grids. 5. Apply learned skills to navigate different themed 5x5 grids, including letters, animals, and sports.	1. Understand and follow simple commands in the 'Robot Statues' game, demonstrating their ability to respond to instructions. 2. Recall the principles of navigating a 4x4 grid, reinforcing their previous learning. 3. Identify the differences between a 4x4 and a 5x5 grid, enhancing their spatial awareness. 4. Successfully navigate a character on a 5x5 grid, applying their understanding of directions. 5. Demonstrate their learning by completing tasks on different themed 5x5 grids (letters, animals, sports), showing adaptability and problem-solving skills.

Module: Creative Stories

This module introduces students to the basics of coding through Scratch Junior, using engaging themes and characters. Each week, students will learn new skills, from basic motion blocks to advanced coding tricks. Encourage creativity and exploration during free play time. For early finishers, provide additional challenges to keep them engaged. Remember to recap previous lessons to reinforce learning.

Duration	Equipment
3 weeks	Required Equipment: • iPad/Tablet
Module Goals	**Module Outcomes**
1. Master the basics of Scratch Junior, including creating a project, exploring the interface, and using 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, enhancing their interaction within the Scratch Junior environment. 4. Gain proficiency in controlling character size and movement, including complex tasks like programming a car to drive across a city background. 5. Understand and apply concepts of character speed control and command sequencing to create interactive scenarios like a race. 6. Develop advanced programming skills such as looping and repeating actions, and apply these to create complex character movements.	1. Master the use of basic motion blocks to move characters in Scratch Junior. 2. Create unique dance routines and change backgrounds in Scratch Junior. 3. Add a second character and make characters walk together in Scratch Junior. 4. Programme a car to drive across a city background and resize it in Scratch Junior. 5. Control character speed, sequence commands, and create an animal race in Scratch Junior. 6. Programme a spaceman to float in space, repeat actions, and create a 'space dance' in Scratch Junior.

Week 1

Lesson: On the Move: Getting Characters in Action

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

In this step-by-step lesson, students will be introduced to Scratch the Cat from Scratch Junior. They will learn how to create a project, explore the interface, and use basic motion blocks to move Scratch. The lesson concludes with a challenge and free play exploration time.

Required equipment for this lesson:

- iPad/Tablet

Learning Goals	Learning Outcomes
• Understand the basic concept and purpose of the Scratch Junior application. • Learn how to create a new project in Scratch Junior. • Identify and understand the functions of the main areas in a Scratch Junior project: the Stage Area, Blocks Toolbox, and Code Area. • Learn how to use basic motion blocks to control the movements of Scratch the Cat. • Apply the knowledge of motion blocks to complete a challenge of moving Scratch the Cat across the screen. • Develop creativity and problem-solving skills through free play and exploration of the Scratch Junior application.	1. Identify and describe the main features of the Scratch Junior interface. 2. Create a new project in Scratch Junior. 3. Understand and apply the function of basic motion blocks in Scratch Junior. 4. Program Scratch the Cat to move across the screen using a sequence of instructions. 5. Experiment with different blocks and sequences to control Scratch the Cat's movements.

Week 2

Lesson: Dance Party: Make Your Characters Groove

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

In this interactive lesson, you'll revisit the basics of Scratch the Cat, learn to change backgrounds in Scratch Junior, and create a unique dance routine for Scratch. If you finish early, you're encouraged to explore the app and experiment with new routines and backgrounds.

Required equipment for this lesson:

- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand and apply the concept of changing backgrounds in Scratch Junior. 2. Develop a dance routine for Scratch the Cat using motion blocks. 3. Experiment with different motions and sequences to create a unique dance. 4. Explore and experiment with the app independently during free play. 5. Enhance creativity and deepen understanding of Scratch Junior through open-ended tasks.	1. Recall and apply knowledge from the 'Scratch The Cat' lesson. 2. Change the background of a project in Scratch Junior. 3. Create a dance routine for Scratch the Cat using motion blocks. 4. Experiment with different motions and sequences to create a unique dance. 5. Explore and experiment with the Scratch Junior app during free play.

Week 3

Lesson: Double the Fun: Adding a Friend for a Walk

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

This step-by-step lesson guides Junior Infants through using Scratch Jr. They will recap on previous knowledge, learn to add a second character, complete a challenge of making characters walk together, learn a trick for advanced coding, and have free play time.

Required equipment for this lesson:

- iPad/Tablet

Learning Goals	Learning Outcomes
• Students will understand and recap the basic functionalities of Scratch Jr, including how to add motion blocks and change backgrounds. • Students will learn how to add a second character to their Scratch Jr project and understand the importance of coding each character separately. • Students will be able to create a sequence of movement blocks for each character, ensuring they move in sync. • Advanced students will learn how to optimize their code by making Scratch the Cat move multiple steps with just one block. • Students will develop problem-solving skills by experimenting with different numbers and movements to make the characters walk smoothly together. • Students will foster creativity and exploration by creating new adventures for Scratch and his friend during free play.	1. Recall and apply the use of motion blocks and changing backgrounds in Scratch Jr. 2. Add a new character to a Scratch Jr project and code it separately. 3. Create a sequence of movement blocks for two characters, ensuring they move in sync. 4. Experiment with different numbers and movements to make characters walk smoothly together. 5. Explore the Scratch Jr app independently, adding more characters or changing the background.

Week 4

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 5

Lesson: Ready, Set, Race! Programming Speedy Sprites

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

In this lesson, students will explore programming through Scratch Jr., creating their own animal race. They'll learn to control character speed and sequence commands. Start by guiding them to add two animal sprites and select a race background. Then, instruct them on programming movements, adjusting character speed, and initiating the race. Finally, encourage creativity in a free play session where they can experiment with different sprites and predict race outcomes.

Required equipment for this lesson:

- iPad/Tablet

Learning Goals	Learning Outcomes
1. Understand and apply basic programming concepts using Scratch Jr. 2. Manipulate sprites by adding, removing, and controlling their speed. 3. Apply sequencing skills to create a desired outcome in a digital project. 4. Develop critical thinking by predicting and observing the outcome of the programmed race. 5. Explore creativity by designing and programming a unique race with different sprites.	1. Understand and apply the process of creating a new project in Scratch Jr. 2. Select, add, and position sprites from the Scratch Jr. library. 3. Choose and set a suitable background for a project. 4. Programme sprites to move across the screen using movement blocks. 5. Adjust the speed of sprites using the speed block to create a competitive race.

Week 6

Lesson: Floating in Space: Looping with the Spaceman

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

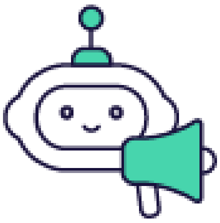
This lesson guides students through a space adventure using Scratch Jr. They'll learn to programme a spaceman to float in space, introducing the concept of repeating actions with the repeat block. The lesson starts with a discussion on space movement, followed by practical steps in Scratch Jr. to create 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.

Module: Exploring Robotics and Commands



This module involves guiding students through the exploration of robotics using Bee-Bots. The lessons are designed to progressively build students' understanding of commands and sequencing. Encourage hands-on participation and group activities to foster problem-solving and collaborative skills. Remember to recap previous lessons to reinforce learning. The module concludes with students applying their skills to trace digital numbers, reinforcing number recognition and programming.

Duration	Equipment
3 weeks	Required Equipment: • Bee-Bots
Module Goals	**Module Outcomes**
1. Master the operation and programming of Bee-Bot robots. 2. Develop understanding of directional commands and sequencing. 3. Apply logical thinking and problem-solving skills in coding activities. 4. Enhance spatial awareness and collaborative skills through group activities. 5. Reinforce number recognition and sequencing skills in the context of programming.	1. Master the operation of Bee-Bot, including forward and backward movements. 2. Understand and execute lateral movements with Bee-Bot, including turns and changes in direction. 3. Apply directional commands and sequencing to perform tasks such as planting flowers using Bee-Bot. 4. Enhance programming skills and understanding of sequencing through activities like guiding Bee-Bot to a specific location. 5. Develop spatial awareness and problem-solving skills by programming Bee-Bot to draw shapes and trace numbers.

Week 1

Lesson: Bee-Bot Basics: Forward and Backward Fun!

□ Beginner	□ 25 mins	System.Threading.Tasks.Task`1[System.String]

In this lesson, you will guide your students in exploring the Bee-Bot, a programmable robot. The lesson involves introducing the Bee-Bot, conducting number strip and colour-based activities, demonstrating Bee-Bot operations, and facilitating student practice. The students will learn to give commands, clear the code, and engage in a group activity. The lesson concludes with a wrap-up reinforcing the skills learned.

Required equipment for this lesson:

- Bee-Bots

Learning Goals	Learning Outcomes
1. Understand the basic functions and controls of a Bee-Bot. 2. Develop skills to instruct Bee-Bot to move forward and backward using a number line. 3. Apply knowledge of Bee-Bot controls in a game-based activity. 4. Comprehend the importance of 'clearing the code' before entering new instructions. 5. Work collaboratively in small groups to guide Bee-Bot along a number line.	1. Identify and describe the basic functions of a Bee-Bot. 2. Successfully instruct a Bee-Bot to move forward and backward using the appropriate buttons. 3. Apply the concept of 'clearing the code' before entering new instructions to a Bee-Bot. 4. Participate in direction games, demonstrating understanding of how to give instructions to a Bee-Bot. 5. Work collaboratively in small groups to guide a Bee-Bot along a number line.

Week 2

Lesson: Twist and Turn: Bee-Bot’s Lateral Moves

□ Beginner	□ 25 mins	System.Threading.Tasks.Task`1[System.String]

In this lesson, you'll guide your students in mastering the art of turning Beebots. You'll recap straight line movements, demonstrate lateral movements, and allow students to practice these skills. The lesson concludes with a review and discussion of the day's learning. Remember, turning is key to navigating Beebots around corners and changing direction.

Required equipment for this lesson:

- Bee-Bots

Learning Goals	Learning Outcomes
1. Understand and apply the concept of turning in Beebot navigation. 2. Differentiate between turning and moving actions in Beebot operation. 3. Execute sequences of turn and move commands to guide Beebot to specific locations. 4. Develop spatial awareness and precise programming skills through hands-on Beebot practice. 5. Engage in reflective discussion to reinforce learning and identify areas for improvement.	1. Understand and demonstrate the difference between turning and moving the Bee-Bot. 2. Successfully navigate the Bee-Bot to turn left and right using the appropriate buttons. 3. Combine turning and moving commands to guide the Bee-Bot in desired directions. 4. Apply turning and moving commands to reach specific points on a grid. 5. Reflect on the challenges and successes experienced during the lesson, reinforcing understanding of Bee-Bot navigation.

Week 3

Lesson: Flower Power: Planting with Bee-Bots

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

In this lesson, you'll guide students through a series of activities to reinforce their understanding of directional commands and sequencing. They'll start by acting as Bee-Bots, then practice programming on an interactive whiteboard, and finally, use real Bee-Bots to plant flowers on a grid mat. The activities are designed to provide hands-on experience with logical thinking and problem-solving processes essential for coding.

Required equipment for this lesson:

- Bee-Bots

Learning Goals	Learning Outcomes
1. Develop understanding and application of directional commands in a fun and interactive manner. 2. Enhance problem-solving skills through the use of Bee-Bots and directional cards. 3. Improve logical thinking by programming Bee-Bots to follow specific sequences. 4. Gain hands-on experience in coding through the 'Planting Flowers' game. 5. Reinforce the concept of sequencing and its importance in coding.	1. Demonstrate understanding of Bee-Bot movements through physical imitation. 2. Apply Bee-Bot programming commands in a digital practice session. 3. Utilise Bee-Bot commands to navigate a grid and 'plant flowers' in a hands-on game. 4. Interpret and follow specific directional commands using Bee-Bot Directional Cards. 5. Reflect on the logical thinking and problem-solving processes used during the lesson.

Week 4

Lesson: Bee-Bot's Bloom Hunt: Finding Flowers

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

In this lesson, you'll guide your students through a series of engaging Bee-Bot activities. They'll act as Bee-Bots, plan and input commands, and work in groups to guide a Bee-Bot to a flower on a grid. This lesson aims to enhance their programming skills and understanding of sequencing.

Required equipment for this lesson:

- Bee-Bots

Learning Goals	Learning Outcomes
1. Develop problem-solving skills through planning and sequencing instructions for Bee-Bot. 2. Understand and apply the concept of programming by inputting planned commands into Bee-Bot. 3. Enhance teamwork and communication skills by collaborating in small groups to guide Bee-Bot to a target. 4. Gain confidence in programming through iterative testing and adjustment of commands. 5. Appreciate the importance of precise instructions and logical sequencing in coding.	1. Demonstrate understanding of programming by acting as Bee-Bots and following commands. 2. Plan and map out routes for Bee-Bot using a whiteboard, demonstrating problem-solving skills and understanding of sequencing. 3. Guide a LadyBird-Bot to find animals on a grid using correct commands in an interactive whiteboard game. 4. Work in small groups to guide a Bee-Bot to a flower on a grid mat, demonstrating understanding of command sequences and testing them. 5. Reflect on the importance of precise commands and correct sequencing in programming during wrap up discussion.

Week 5

Lesson: Square Dancing: Drawing Shapes with Bee-Bot

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

In this lesson, you'll guide your students in understanding directional commands and spatial awareness through programming Bee-Bots to move in square shapes. They'll play a game, draw squares, trace numbers, and work in groups to program Bee-Bots. This will enhance their sequencing, problem-solving, and collaborative skills, setting a strong foundation for advanced coding concepts.

Required equipment for this lesson:

- Bee-Bots

Learning Goals	Learning Outcomes
1. Develop understanding of sequencing and spatial awareness through programming Bee-Bots to move in square shapes. 2. Enhance precision and planning skills in programming by creating both small and large squares. 3. Improve problem-solving abilities by adjusting the number of forward steps between each turn to create larger squares. 4. Strengthen collaborative skills through group work in programming Bee-Bots. 5. Build a strong foundation in logical thinking and programming for more advanced coding concepts in the future.	1. Master the use of directional commands to program Bee-Bots to move in square shapes. 2. Develop an understanding of sequencing and spatial awareness through the programming of Bee-Bots. 3. Understand the importance of precision and planning in programming by creating both small and large squares. 4. Enhance problem-solving and collaborative skills by working in groups to program Bee-Bots. 5. Prepare for more advanced coding concepts by building a strong foundation in logical thinking and programming.

Week 6

Lesson: Number Tracer: Bee-Bot’s Digital Path

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

In this lesson, you'll guide students through programming Bee-Bots to trace digital numbers. Starting with a slideshow of digital numbers, students will then plan Bee-Bot paths on whiteboards before practising with an interactive game. Finally, they'll use Bee-Bots to trace numbers on a mat, reinforcing number recognition, sequencing skills, and programming.

Required equipment for this lesson:

- Bee-Bots

Learning Goals	Learning Outcomes
1. Develop understanding and recognition of digital number shapes. 2. Enhance critical thinking through planning and sequencing of Bee-Bot commands. 3. Improve ability to execute planned commands using Bee-Bots. 4. Engage in interactive learning through digital tracing practice. 5. Apply knowledge of digital numbers and programming in a challenging extension activity.	1. Identify and differentiate between standard and digital number formats. 2. Plan and sequence commands for Bee-Bot to trace digital numbers. 3. Execute the planned commands using Bee-Bot on a digital platform. 4. Apply critical thinking to program Bee-Bot to trace numbers on a physical mat. 5. Engage in peer learning through an extension activity, demonstrating understanding of number shapes and programming process.