Subject: Computing – Programming – Moving a Robot   Year: A – phase 1 –Unit 2/5

NC/PoS:
- Understand what algorithms are, how they are implemented as programs on digital devices, and that programs execute by following precise and unambiguous instructions.
- Create and debug simple programs.
- Use logical reasoning to predict the behaviour of simple programs.

Prior Learning (what pupils already know and can do)
- Following simple instructions in daily routines.
- Recognising directions such as forward, backward, left, and right.
- Basic problem-solving and sequencing tasks.

End Points (what pupils MUST know and remember)
- Understand and use directional commands to control a robot.
- Recognise that computers follow precise instructions.
- Create and debug simple sequences of commands.
- Plan and execute a program to navigate a robot to a target.

Key Vocabulary
instruction, command, debug, forward, backward, left, right, program, sequence, robot, route

Recommended Resources:
- Bee-Bots or alternative floor robots.
- Printable grid mats with themed scenarios (e.g., treasure maps, roads).
- Manipulatives like arrows or cards for offline programming.

Unplugged activities provide possible opportunities for the children to record.

Curriculum Connections:
- Maths: Understanding direction, position, and turns (e.g., whole, half, quarter turns).
- Physical Education: Navigating physical spaces using directional commands.
- Literacy: Using precise language for giving instructions.
- Geography: map of the street around school, directional language

Career Opportunities:
- Industrial Robot Operator: Programming and operating robots in factories.
- Warehouse Automation Specialist: Managing robotic systems for packing and shipping.
- Military Drone Operator: Programming and controlling unmanned vehicles.
- STEM Educator: Teaching robotics and programming to future generations.
- Agricultural Engineer: Programming robotic systems for planting, harvesting, and monitoring.

Session 1: Exploring Commands

Objective: Understand what a command is and how it affects the robot.
Digital Activity: Experiment with the buttons on Bee-Bots to observe outcomes.
Unplugged Activity: Use printed arrows to simulate commands with peers acting as "robots."

Key Vocabulary: command, forward, backward

Session 2: Giving Directions

Objective: Use directional language to guide movement.
Digital Activity: Guide Bee-Bots along a simple path on a grid mat using directional commands.
Unplugged Activity: Play a game where children give step-by-step instructions to guide a classmate to a target spot in the room.

Key Vocabulary: left, right, forward, backward

Session 3: Programming Forward and Backward

Objective: Create simple programs using forward and backward commands.
Digital Activity: Use Bee-Bots to navigate a straight path to a target.
Unplugged Activity: Arrange cards with "forward" and "backward" commands in sequence to guide a toy car.

Key Vocabulary: sequence, forward, backward, program

Session 4: Combining Commands

Objective: Combine forward, backward, left, and right commands to navigate a path.
Digital Activity: Program Bee-Bots to move around obstacles to reach a target.
Unplugged Activity: On a large grid mat, children walk a path by following a sequence of commands provided by a partner.

Key Vocabulary: left, right, sequence, direct

Session 5: Debugging Programs

Objective: Identify and correct mistakes in a sequence of commands.
Digital Activity: Run a pre-written Bee-Bot program, observe unexpected outcomes, and debug the sequence.
Unplugged Activity: Use printed arrows to fix an incorrect sequence provided on a worksheet.

Key Vocabulary: debug, error, correct

Session 6: Creating Routes

Objective: Plan and program a route for the robot, considering multiple solutions.
Digital Activity: Design and test a route on a themed mat (e.g., guiding the robot to a treasure chest).
Unplugged Activity: Draw routes on grid paper and write corresponding commands; peers test the routes using manipulatives.

Key Vocabulary: route, plan, solution

Future learning this content supports:
Progression to more complex algorithms and debugging.
Introduction to loops and conditional statements in higher years.

May 25