Subject: Computing

Key Concept/ Theme: Programming. Year 5 unit – Micro:Bit.

In this unit, pupils will develop their understanding of programming by using the Micro:bit. They will learn how to create algorithms, use variables, work with inputs/outputs, and debug effectively. Pupils will explore how physical computing interacts with the real world through purposeful design challenges.

Skills and Knowledge for subject units:

Key Skills:

- ✓ Decomposing a program without support.
- ✓ Predicting how software will work based on previous experience.
- ✓ Writing more complex algorithms for a purpose.
- ✓ Programming an animation.
- ✓ Iterating and developing their programming as they work.
- ✓ Confidently using loops in their programming.
- ✓ Using a more systematic approach to debugging code, justifying what is wrong and how it can be corrected.
- ✓ Writing code to create a desired effect.
- ✓ Using a range of programming commands.
- ✓ Using repetition within a program.
- ✓ Using logical thinking to explore software more independently, making predictions based on their previous experience.
- ✓ Identify ways to improve and edit programs, videos, images etc.

Knowledge:

- ✓ To know that a Micro:bit is a programmable device.
- ✓ To know that Micro:bit uses a block coding language similar to Scratch.
- ✓ To understand and recognise coding structures including variables.
- ✓ To know what techniques to use to create a program for a specific purpose (including decomposition).

Vocabulary for unit: algorithm, animation, app, blocks, bluetooth, code block, connection, create, debug, decompose, designing, desktop, device, download, images, input, instructions, laptop, load, loop, Micro:bit, outputs, pairing, pedometer, polling, predict, repetition, reset, sabotage, scoreboard, screen, systematic, tablet, tinkering, USB, variables, wi-fi, wireless, wires	
Federation/school specific areas to cover (Add in any local areas of study, trips and people)	
1.	**Prior learning reconnection (year group, cycle & term):** Basic understanding of input/output devices and block-based programming e.g. using 2Code (PM) or Scratch. Y4, Cycle A, Term 2&5 **Cross – curricular links:** Science – understanding how inputs are similar to real-world triggers like light sensors or switches. DT – exploring how devices are constructed and the purpose of user interaction. **LO: To tinker with a new piece of software (Micro:bit).** *Activity ideas to achieve the LO:* Introduction to the Micro:bit. Pupils explore the interface and simulate basic outputs (e.g., LED display). **End point:** Pupils can identify basic components of a Microbit and create a simple programme using inputs (e.g., button A to show a message). **Vocabulary focus for this lesson:** Microcontroller, LED, Input, Output **Possible misconceptions to consider:** - Pupils may think MicroBits work without any programming. - Confusing input devices with output devices (e.g., believing the LED display is an input). - Believing button A and B do the same thing by default.
2.	Prior learning reconnection: Familiarity with co-ordinates, block coding to manipulate visuals, simple loops (Scratch, Y4, Cycle A, Terms 2 & 5) **Cross – curricular links:** Maths (Coordinates): LED grid introduces basic x/y plotting concepts. Art: Using patterns and pixel art to design animations or icons. **LO: To program an animation.** *Activity ideas to achieve the LO:* Use MakeCode to create basic animations or messages on the Micro:bit LED grid. **End point:** Pupils can program the LED matrix to show custom icons or animations in response to inputs. **Vocabulary focus for this lesson:** Algorithm, Sequence, Loop **Possible misconceptions to consider:** - Pupils may expect smooth images rather than pixel-based designs. - Assuming the LED grid is colour-capable.

	- Confusing the orientation of the grid (mixing up rows and columns).
3.	Prior learning reconnection: Pupils may have explored 'if statements' in advanced programming – Scratch, Y4, Term 5, Cycle A **Cross – curricular links:** Maths (Algebra): Understanding variables as symbolic placeholders for changing values. PE: Linking to scoring or reaction-based games. **LO: To recognise coding structures.** *Activity ideas to achieve the LO:* Design a program that responds to button presses (e.g., counter or reaction game). **End point:** Pupils create a simple program that uses variables to track data (e.g., a counter for how many times button A is pressed). **Vocabulary focus for this lesson:** Input, Condition, Event **Possible misconceptions to consider:** - Thinking variables are permanently fixed after assignment. - Using multiple variables when only one is needed. - Expecting the variable to display automatically without a show command.
4.	Prior learning reconnection: use of Loops in Scratch (Y4, Cycle A, Term 5) **Cross – curricular links:** PSHE (Fair play, decision-making): Using chance-based games. Maths (Probability): Exploring randomness and fair tests. **LO:** To create a program for a specific task *Activity ideas to achieve the LO:* Create a step counter or temperature display. Focus on introducing and using variables. **End point:** Pupils program a game using conditional logic and random number generation.

	Vocabulary focus for this lesson: Variable, Data, Logic **Possible misconceptions to consider:** - Confusing `if` and `else` blocks. - Expecting truly random outcomes each time without understanding limitations of pseudo-randomness. - Not resetting variables when replaying the game.
5.	Prior learning reconnection **Cross – curricular links:** Science (Sound): Understanding how vibrations and frequencies create sound. Music: Using sound creatively, pitch variation and tempo. **LO:** To create a program. *Activity ideas to achieve the LO:* Pupils plan a mini-project (e.g., digital pet, timer) and present it to the class. **End point:** Pupils incorporate sound or attach an external component (e.g., buzzer or servo) and control it via code. **Vocabulary focus for this lesson:** Evaluation, Design, Purpose **Possible misconceptions to consider:** - Believing sound will play from the MicroBit without a speaker/buzzer. - Confusing analogue and digital signals. - Miswiring external components, especially if using crocodile clips or breadboards.
6.	Assessment – see quiz and knowledge capture

Things to note:

For a 6 week term 4 lessons and an assessment

For a 7/8 week terms 5 lessons and an assessment

The progression of skills and knowledge need to be thought about so that they are covered by the teaching and learning this term

Adaptions:

Examples could be:

- resources (technology, physical/concrete)
- Location of the lesson
- Groupings/.staffing support
- SEND/Inclusion

Summative End Points: **Which NC statements from the overview document are expected to be achieved or specifically developed?**

Pupils should be taught to:

- Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.
- Use sequence, selection, and repetition in programs; work with variables and various forms of input and output.
- Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.