



**Declarative, Procedural and Disciplinary Knowledge and Skill Tracker**



**Computing Coding**

|   | Year 1  | Year 2   | Year 3   | Year 4   | Year 5   | Year 5   |
|---|---|--|--|--|--|--|
| <p><b>Declarative Knowledge</b><br/>(Declarative Knowledge is explicit knowledge about facts, histories, ideas, topics, principles, and concepts. )</p> | <p>Understand what an algorithm is and be able to create a simple algorithm.</p> <p>Begin to predict what will happen for a short sequence of instructions.</p> <p>Begin to use different software or applications to create movement and patterns On a screen.</p> | <p>Understand what an algorithm is and demonstrate simple linear algorithms.</p> <p>Use logical reasoning to predict and debug more complex programs</p> | <p>Understand how an algorithm is implemented using a sequence of precise instructions.</p> <p>Can predict the outcome of a Sequence of precise instructions.</p> <p>Detect a problem in an algorithm, which could result in a different outcome to the one intended.</p> <p>Understand what inputs and outputs are, how they can be used.</p> <p>Designs, writes, executes and debugs programs of increasing complexity that Accomplish a specific goal.</p> <p>Use logical reasoning to predict and debug more complex programs including inputs and outputs</p> | <p>Can solve problems by debugging errors.</p> <p>Understands how to decompose a problem into smaller parts with some verbal reasoning.</p> <p>Understands how sequencing, using inputs and repetition in programs has specific effects on the output.</p> <p>Recognise that an algorithm will help to sequence more complex programs. Use logical reasoning to predict more complex programs including loops and repeats.</p> | <p>Understands how to decompose more open-ended problems into smaller parts, provides some Reasoning for their choices.</p> <p>Approaches a range of problems using computationally thinking concepts, helping them to design other algorithms for other specific outcomes.</p> <p>Use logical reasoning to predict and debug more complex programs including selection.</p> | <p>Understand the importance of planning, testing and correcting algorithms.</p> <p>Demonstrate a range of different strategies to solve a problem including: abstraction, decomposition, logic &amp; evaluation.</p> <p>Understand why sequence &amp; Patterns are important when creating simple algorithms that Are part of a more complex program.</p> <p>Gives reasoning for each step Within algorithms and applying Them to a program.</p> <p>Understand &amp; develop flow diagrams.</p> |

|   |   |  |  |  |  |   |
|---|---|--|--|--|--|---|
| <p><b>Procedural Knowledge</b><br/><b>(It can be simply stated as knowing how to do something.)</b></p> | <p>Can explain how algorithms are used in everyday life.</p> <p>Knows how to give instructions to a friend and follow their instructions to move around a space.</p> <p>Describe what happens when buttons are pressed on a robot or device.</p> <p>Knows how to press buttons in the correct order to make a robot follow a short sequence.</p> <p>knows how to use the word debug to correct an algorithm that doesn't work in the way it was intended.</p> | <p>Knows how to programme a robot or app to do a particular task.</p> <p>Explains the order needed to do things to make something happen and to talk about it as an algorithm.</p> <p>Uses programming applications to make objects move. Creates and debugs with improved confidence &amp; efficiency.</p> <p>Beginning to program using simple block code.</p> | <p>Can design and program a simple program.</p> <p>Knows how to repeatedly test a program and recognise when to debug .</p> <p>Can provide examples of how to use inputs and outputs effectively</p> | <p>Design simple algorithms using loops and repeats, whilst detecting and correcting errors is debugging.</p> <p>Write and execute an efficient program, using loops such as forever, repeat &amp; repeat until commands.</p> <p>Can use 'loops' and understands their effect.</p> | <p>Program a condition that uses a sensor to detect a change, which can select an action within a program.</p> <p>Design, write and execute an Efficient program, including selection (IF...THEN) command.</p> <p>Change an input to a program to achieve a different output.</p> <p>Uses programs linked to physical systems and sensors e.g. the alarm goes off when the sensor is triggered.</p> <p>Design, write and execute an Efficient program, which demonstrates and understanding of the difference between, and appropriate use of IF...THEN, IF...THEN...ELSE, And nested IF statements.</p> | <p>Use a variable to increase programming possibilities.</p> <p>Use a variable and relational Operators (e.g. &lt; =&gt;) within a loop to stop a program.</p> <p>Evaluate the effectiveness and efficiency of an algorithm while Continually testing the programming of that program.</p> <p>Use different inputs (including sensors) to control a device or Onscreen action and predict what will happen.</p> <p>Use logical reasoning to predict And debug more complex programs including: selection.</p> |
|---|---|--|--|--|--|---|

|  |   |  |  |   |   |   |
|--|---|--|--|---|---|---|
| <p><b>Disciplinary Knowledge</b></p> <p>(This is best described as the action taken within a particular subject to gain knowledge i.e. how we gain substantive knowledge. For example, in history this might mean using evidence to construct a claim. Meanwhile, in science it might mean testing hypotheses. In music, it might mean reading and writing notation. As you can see, it really is quite distinct within each domain. )</p> | <p>Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions</p> | <p>Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions</p> <p>Create and debug simple programs</p> <p>Use logical reasoning to predict the behaviour of simple programs</p> | <p>design, write and debug programs that accomplish specific goals.</p> <p>Use sequence, selection in programs</p> | <p>design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems.</p> <p>Use sequence, selection, and repetition in programs.</p> | <p>design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts</p> <p>Use sequence, selection, and repetition in programs; work with variables and various forms of input and output</p> | <p>design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts</p> <p>use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs</p> |
|--|---|--|--|---|---|---|