**National Curriculum:**

The national curriculum for computing aims to ensure that all pupils:

* can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation
* can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems
* can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems
* are responsible, competent, confident and creative users of information and communication technology.

**Key stage 1**

**Pupils should be taught to:**

* 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
* use technology purposefully to create, organise, store, manipulate and retrieve digital content
* recognise common uses of information technology beyond school
* use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies.

**Key stage 2**

**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
* understand computer networks including the internet; how they can provide multiple services, such as the world wide web; and the opportunities they offer for communication and collaboration
* use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content
* select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information
* use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.

**EYFS**

|  |  |
| --- | --- |
| **Vocabulary**  **By the end of EYFS they will be able touse the words:** | **Outcomes for the end of EYFS. Children will be able to:** |
| App  Computer  Device  Information  Interactive Whiteboard  Internet  iPad  Keyboard  Laptop  Mobile phone  Mouse  Online Safety  Program  Tablet  Technology | **Fine Motor Skills**  Develop their fine motor skills so that they can use a range of tools competently, safely and confidently  **Managing Self**  Be confident to try new activities and show independence, resilience and perseverance in the face of challenge  Explain the reasons for rules, know right from wrong and try to behave accordingly.  **Understanding the World**  Know some similarities and differences between things in the past and now, drawing on their experiences and what has been read in class. |

**Cycle A, Autumn Term 1**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **EYFS & Year 1** | **Lesson 1** | **Lesson 2** | **Lesson 3** | **Lesson 4** | **Lesson 5** | **Lesson 6** |
| **Online Safety**  Rule writers  (yr1 SoC)  **Computer Science**  **Course B** | Rule writers (yr1 SoC)  To develop a simple set of age-appropriate rules to establish a working framework for online safety for school and home during Year EYFS/Yr1 | Code.org -L2  Move it, Move it  I know how to define a list of steps (algorithm) to get a friend from their starting position to their goal  I know how to identify and fix errors in the execution of an algorithm  I know how to translate a list of steps into a series of physical actions | Code.org -L3  Sequencing  I know how to experiment with standard block-based programming actions such as: clicking, drag and drop, etc.  I know how to model proper computer lab behaviors | Code.org -L4  Programming  I know how to build a computer program from a set of written instructions.  I know how to construct a program by reorganizing sequential movements.  I know how to translate movements into a series of commands. | Code.org -L5  Programming  I know how to identify and locate bugs in a program.  I know how to translate movements into a series of commands. | Code.org -L6  Getting Loopy  I know how to convert a series of multiple actions into a single loop.  I know how to repeat actions initiated by the instructor.  I know how to translate a picture program into a real-world dance. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Year 2 & Year 3** | **Lesson 1** | **Lesson 2** | **Lesson 3** | **Lesson 4** | **Lesson 5** | **Lesson 6** |
| **Online Safety**  Rule writers  (yr2 SoC)  **Computer Science**  **Course D** | Rule writers (yr2 SoC)  To develop a simple set of age-appropriate rules to establish a working framework for online safety for school and home during Year 2&3 | Code.org -L2  Graph Paper Programming  I know how to explain constraints of translating problems from human language to machine language  I know how to reframe a sequence of steps as an encoded program | Code.org -L3  Online Puzzles  I know how to break down a long sequence of instructions into the largest repeatable sequence.  I know how to modify an existing program to solve errors.  I know how to order movement commands as sequential steps in a program. | Code.org -L4  Relay Programming  I know how to define ideas using code and symbols.  I know how to identify signs of frustration  I know how to verify work done by teammates. | Code.org -L5  Debugging  I know how to describe and implement a plan to debug a program.  I know how to identify a bug and the problems it causes in a program.  I know how to read and comprehend given code. | Code.org -L6  Events in Bounce  I know how to create an interactive game using sequence and event-handlers.  I know how to identify actions that correlate to input events.  I know how to share a creative artifact with other students. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Year 4, Year 5 & Year 6** | **Lesson 1** | **Lesson 2** | **Lesson 3** | **Lesson 4** | **Lesson 5** | **Lesson 6** |
| **Online Safety**  **Age limits**  **Computer Science**  **Course F** | Online Safety  - Age Limits  What are the different age limits for popular apps.  Why do these have age limits?  What dangers do users face on these apps?  How can you stay safe if using these app?  Scenarios given and discussed | Computer Science – Code.org  L2  Introducing Sprite Lab | Computer Science – Code.org  L3  Making Sprites  I know how to make a sprite and change its size  I know about the two concepts at the heart of Sprite Lab: Sprites and behaviours | Computer Science – Code.org  L4  Sprites in action  I know how to add a time event.  I know how to add a key event  I know how to switch behaviours | Computer Science – Code.org  L5  Mini project – Virtual pet  I know how to create an interactive pet that looks and behaviours how I wish it to.  I know how to customize appearance | Computer Science – Code.org  L6  Blank Space Stories (Unplugged activity) |