Year 1

Skills

Hardware

- Learning how to operate a camera to take photos and videos.
- Learning how to explore and tinker with hardware to find out how it works.
- Learning where keys are located on the keyboard.

Computational thinking

- Learning that decomposition means breaking a problem down into smaller parts.
- Using decomposition to solve unplugged challenges.
- Using logical reasoning to predict the behaviour of simple programs.
- Developing the skills associated with sequencing in unplugged activities.
- Following a basic set of instructions.
- Assembling instructions into a simple algorithm.

Programming

- Programming a floor robot to follow a planned route.
- Learning to debug instructions when things go wrong.
- Learning to debug an algorithm in an unplugged scenario.
- Recognising that robots are programmed by humans.
- Explaining what they are trying to achieve with their algorithms.
- Writing clear, sequenced algorithms for familiar tasks
- Using terms like ‘start,’ ‘end’ and ‘next’ to describe the steps in algorithms.
- Changing their instructions or algorithms into code that the robot understands.
- Beginning to identify errors in algorithms.
- Making suggestions for how to fix errors in algorithms.

Knowledge

Programming

- To understand that an algorithm is when instructions are put in an exact order
- To understand that decomposition means breaking a problem into manageable chunks and that it is important in computing.
- To know that we call errors in an algorithm 'bugs' and fixing these 'debugging'.
- To understand the basic functions of a Bee-Bot.
- To know that you can use a camera/tablet to make simple videos
- To know that algorithms move a bee-bot accurately to a chosen destination.
- To know humans need to give robots instructions to follow and that they will carry out these instructions exactly, even if they are wrong.

Year 2

Skills

Hardware

- Understanding what a computer is and that it’s made up of different components.

- Recognising that buttons cause effects and that technology follows instructions.
- Learning how we know that technology is doing what we want it to do via its output.
- Developing confidence with the keyboard and the basics of touch typing.

Computational thinking

- Articulating what decomposition is
- Decomposing a game to predict the algorithms used to create it.
- Learning that there are different levels of abstraction.
- Explaining what an algorithm is.
- Following an algorithm.
- Creating a clear and precise algorithm.

Programming

- Using logical thinking to explore software, predicting, testing and explaining what it does.
- Using an algorithm to write a basic computer program.

Knowledge

Computing systems and networks

- To know that computers often work together.

Programming

- To understand what machine learning is and how that enables computers to make predictions.
- To know that abstraction is the removing of unnecessary detail to help solve a problem.
- To know that coding is writing in a special language so that the computer understands what to do
- To understand that the character in ScratchJr is controlled by the programming blocks.
- To know that you can write a program to create a musical instrument or tell a joke.
- To know that programming a computer or device involves giving it instructions to perform specific tasks
- That video games, phones, websites and apps are all created using programming.
- To know that different devices and programs use different programming languages or 'codes'
- To know that an algorithm becomes a program when it is coded.
- That programs execute the exact instructions they are given, even if they are incorrect.
- That a program is a series of instructions (algorithms) that are written for a computer to follow.
- That a person can program a device by giving it an algorithm/algorithms to follow.

Data handling

- To understand what steps you need to take to create an algorithm.

National curriculum — end of KS1

- 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.

KS2

Year 3

Skills

Hardware

- Understanding what the different components of a computer do and how they work together.
- Learning about the purpose of routers.
- Drawing comparisons across different types of computers.

Networks and data representation

- Understanding the role of the key components of a network.
- Identifying the key components within a network, including whether they are wired or wireless.
- Understanding that websites & videos are files that are shared from one computer to another.
- Learning about the role of packets.
- Understanding how networks work and their purpose.
- Recognising links between networks and the internet.
- Learning how data is transferred.

Computational thinking

- Using decomposition to explain the parts of a laptop computer
- Using decomposition to explore the code behind an animation.
- Using repetition in programs.
- Using logical reasoning to explain how simple algorithms work.
- Explaining the purpose of an algorithm.
- Forming algorithms independently.

Programming

- Making reasonable suggestions for how to debug their own and others' code.
- Continuing existing code.
- Incorporating loops to make code more efficient.
- Using logical thinking to explore more complex software; predicting, testing and explaining what it does.

Knowledge

Computing systems and networks

- To know the components that make up a network (Wireless access point/WAP, Network switch, Router, Server and devices).
- To know that a server is central to a network and responds to requests made.
- To know that the internet connects all the networks around the world.
- To know what a packet is and why it is important for website data transfer.
- To know the roles that inputs and outputs play on computers.
- To know what some of the different components inside a computer are e.g. CPU, RAM, hard drive, and how they work together.

Programming

- To know that Scratch is a programming language and some of its basic functions.
- To understand how to use loops to improve programming.
- To understand how decomposition is used in programming.
- To understand that you can remix and adapt existing code.

Year 4

Skills

Hardware

- Using tablets or digital cameras to film a weather forecast.
- Understanding that weather stations use sensors to gather and record data which predicts the weather.

Networks and data representation

- Understanding that computer networks provide multiple services, such as the World Wide Web, and opportunities for communication and collaboration.

Computational thinking

- Using decomposition to solve a problem by finding out what code was used.
- Using decomposition to understand the purpose of a script of code.
- Identifying patterns through unplugged activities.
- Using abstraction to identify the important parts when completing both plugged and unplugged activities.
- Using past experiences to help solve new problems.
- Breaking down what they want to achieve into smaller, manageable parts.
- Using logic, pattern recognition and decomposition to solve simple problems.
- Remixing code to alter and add to an existing program.

- Recognising repeating patterns in a program or code.
- Creating loops to make code more efficient in block-based programs.
- Beginning to use variables in block-based programming languages to make programs more interactive.
- Including a conditional statement in block-based programming languages
- Recognising the relationship between what is happening in a program and the written (block) code.

Programming

- Creating algorithms for a specific purpose.
- Coding a simple game.
- Using abstraction and pattern recognition to modify code.
- Incorporating variables to make code more efficient.

Knowledge

Programming

- To understand that a variable is a value that can change (depending on conditions) and know that you can create them in Scratch.
- To know what a conditional statement is in programming.
- To understand that variables can help you to create a quiz on Scratch.
- To know that combining computational thinking skills can help you to solve a problem.
- To understand that pattern recognition means identifying patterns to help them work out how the code works.
- To understand that algorithms can be used for a number of purposes e.g. animation, games design etc

Year 5

Skills

Hardware

- Learning that external devices can be programmed by a separate computer.

Networks and data representation

- Learning the vocabulary associated with data: data and transmit.
- Recognising that computers transfer data in binary and understanding simple binary addition.
- Learning that messages can be sent by binary code, reading binary up to eight characters and carrying out binary calculations.

Computational thinking

- Decomposing animations into a series of images.
- Decomposing a program without support.
- Decomposing a story to be able to plan a program to tell a story.
- Predicting how software will work based on previous experience
- Writing more complex algorithms for a purpose.

Programming

- 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.
- Amending code within a live scenario.
- Decomposing a program independently when given a specific outcome or task to achieve.
- Altering existing code with a new, specific outcome in mind.
- Independently using loops to make code more efficient in text-based programs.
- Using nested loops to make code more efficient.
- Using variables in block-based programming languages and understanding the impact of changing the variables in their code.
- Explaining what a program does and how it works, referring to the inputs and outputs.
- Becoming more efficient and effective at debugging their programs.
- Systematically identify mistakes, problems or 'bugs' in a program.

Knowledge

Computing systems and networks

- To know how search engines work.
- To know the difference between ROM and RAM.

- To know that a soundtrack is music for a film/video and that one way of composing these is on programming software.
- To understand that using loops can make the process of writing music simpler and more effective.
- To know how to adapt their code while performing their music.
- 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).

Data handling

- To know what numbers using binary code look like and be able to identify how messages can be sent in this format.

Year 6

Skills

Hardware

- Learning about the history of computers and how they have evolved over time.
- Understanding and identifying barcodes, QR codes and RFID.
- Identifying devices and applications that can scan or read barcodes, QR codes and RFID.
- Identify different types of AI and their applications in everyday life.

Computational thinking

- Decomposing a program into an algorithm.

- Using past experiences to help solve new problems.
- Writing increasingly complex algorithms for a purpose.
- Analysing the effectiveness of prompts and refine them for improved AI outputs

Programming

- Debugging quickly and effectively to make a program more efficient.
- Remixing existing code to explore a problem.
- Using and adapting nested loops.
- Programming using the language Python.
- Changing a program to personalise it.
- Evaluating code to understand its purpose.
- Predicting code and adapting it to a chosen purpose.
- Applying coding skills like decomposition and pattern recognition to interact with AI applications.
- Recognising a wider range of text-based programming languages.
- Making links between different programming interfaces that they are faced with.
- Recognising examples of programming elements in real-life applications.
- Decomposing a program independently when given a specific outcome or task to achieve.
- Live coding (improvising with code).
- Altering existing code with a new, specific outcome in mind.
- Independently using loops to make code more efficient in text-based or block-based programs.
- Using nested loops to make code more efficient.
- Explaining what a program does and how it works, referring to the inputs and outputs.
- Becoming more efficient and effective at debugging their programs.
- Systematically identifying mistakes, problems or 'bugs' in a program.

Knowledge

Computing systems and networks

- To know that AI is trained on data to recognise patterns and generate outputs.

Programming

- To know that there are text-based programming languages such as Logo and Python.
- To know that nested loops are loops inside of loops.
- To understand the use of random numbers and remix Python code.
- To know code may sometimes need to be downloaded onto a physical system/device.
- To know programmers often save time when creating code by taking code from one program and turning it into another.
- To know nested loops are loops within loops.
- To know it is important to follow the syntax rules in a programming language so that the computer understands what we are trying to tell it, but that we do not need to remember all these rules.
- To know many text-based coding languages use brackets or indentation to show which code belongs to a particular function or loop.
- To know typing and spacing are very important in text-based languages and can cause errors in code if used incorrectly.
- To know running a program to identify errors should be done before checking the code.
- To know errors in a program could be as a result of forgetting to 'end' a loop.

Data handling

- To know that data contained within barcodes and QR codes can be used by computers.
- To know that Radio Frequency Identification (RFID) is a more private way of transmitting data.

- To know that data is often encrypted so that even if it is stolen it is not useful to the thief.

National curriculum — end of KS2

- 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.