Progression of Computing Knowledge and Skills

Strand: **Computer Science**

EYFS	YEAR 1	YEAR 2	YEAR 3	YEAR 4	YEAR 5	YEAR 6
Hardware						
*Learning how to operate a camera to take photographs of meaningful creations or moments • Learning how to explore and tinker	Learning how to explore and tinker with hardware to find out how it works • Understanding that computers and devices around us use inputs	• Understanding what a computer is and that it's made up of different components • Recognising that buttons cause effects and that technology follows instructions	Understanding what the different components of a computer do and how they work together • Drawing comparisons across	Learning about the purpose of routers	Learning that external devices can be programmed by a separate computer • Learning the difference between ROM and RAM	• Learning about the history of computers and how they have evolved over time • Using the understanding of historic computers to

with hardware to develop familiarity and introduce relevant vocabulary • Learning how to operate a camera • Recognising that a range of technology is used in places such as homes and schools • Learning what a keyboard is and how to locate relevant keys • Learning what a mouse is and developing basic mouse skills such as moving and clicking	and outputs, identifying some of these • Learning where keys are located on the keyboard • Learning how to operate a camera	• Learning how we know that technology is doing what we want it to do via its output. • Using greater control when taking photos with tablets or computers • Developing confidence with the keyboard and the basics of touch typing	different types of computers • Learning what a server does		• Recognising how the size of RAM affects the processing of data • Understanding the fetch, decode, execute cycle	design a computer of the future • Understanding and identifying barcodes, QR codes and RFID • Identifying devices and applications that can scan or read barcodes, QR codes and RFID • Acknowledging that corruption can happen within data during transfer (for example when downloading, installing, copying and updating files)
EYFS	**YEAR 1**	**YEAR 2**	**YEAR 3**	**YEAR 4**	**YEAR 5**	**YEAR 6**
Network and Data Representation						
	• Understanding what the internet is		Learning what a network is and its purpose • Identifying the key components within a network, including whether they are wired or wireless • Recognising links between networks and the internet	Consolidating understanding of the key components of a network • Understanding that websites & videos are files that are shared from one computer to another • Learning about the role of packets	Learning the vocabulary associated with data: data and transmit • Learning how the data for digital images can be compressed • Recognising that computers transfer data in binary and	• Understanding that computer networks provide multiple services

			• Learning how data is transferred	• Understanding that computer networks provide multiple services, such as the World Wide Web, and opportunities for communication and collaboration	understanding simple binary addition • Relating binary signals (Boolean) to the simple character-based language, ASCII • Learning that messages can be sent by binary code, reading binary up to 8 characters and carrying out binary calculations • Understanding how bit patterns represent images as pixels	
EYFS	**YEAR 1**	**YEAR 2**	**YEAR 3**	**YEAR 4**	**YEAR 5**	**YEAR 6**
Computational Thinking						
Using logical reasoning to read simple instructions and predict the outcome	• 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	• Articulating what decomposition is • Decomposing a game to predict the algorithms used to create it • Using decomposition to decompose a story into smaller parts • Learning what abstraction is	Using decomposition to explain the parts of a laptop computer • Using decomposition to explore the code behind an animation • Using repetition in programs • Understanding that computers follow instructions	Solving unplugged problems by decomposing them into smaller parts • Using decomposition to understand the purpose of a script of code • Using decomposition to help solve problems • Identifying patterns through unplugged activities	• 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	Decomposing a program into an algorithm • Using past experiences to help solve new problems • Writing increasingly complex algorithms for a purpose

	• Developing the skills associated with sequencing in unplugged activities • Learning that an algorithm is a set of step by step instructions used to carry out a task, in a specific order • Follow a basic set of instructions • Assembling instructions into a simple algorithm	• Learning that there are different levels of abstraction • Explaining what an algorithm is • Following an algorithm • Creating a clear and precise algorithm • Learning that computers use algorithms to make predictions • Learning that programs execute by following precise instructions • Incorporating loops within algorithms	• Using an algorithm to explain the roles of different parts of a computer • Using logical reasoning to explain how simple algorithms work • Explaining the purpose of an algorithm • Forming algorithms independently	• Using past experiences to help solve new problems • Using abstraction to identify the important parts when completing both plugged and unplugged activities • Creating algorithms for a specific purpose	• Writing more complex algorithms for a purpose	
EYFS	**YEAR 1**	**YEAR 2**	**YEAR 3**	**YEAR 4**	**YEAR 5**	**YEAR 6**
Programming						
Following instructions as part of practical activities and games and learning to debug when things go wrong • Learning to give simple instructions • Learning that an algorithm is a set of instructions to carry out a task, in a specific order	Programming a Bee-bot/Virtual Bee-bot to follow a planned route • Learning to debug instructions when things go wrong • Developing a howto video to explain how the Bee-bot works.	Using logical thinking to explore software, predicting, testing and explaining what it does • Using an algorithm to write a basic computer program • Learning what loops are	• Using logical thinking to explore more complex software; predicting, testing and explaining what it does • Incorporating loops to make code more efficient • Remixing existing code	Understanding that websites can be altered by exploring the code beneath the site • Coding a simple game • Using abstraction and pattern recognition to modify code	• Programming an animation • Iterating and developing their programming as they work • Beginning to use nested loops (loops within loops) • Debugging their own code	• 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

• Experimenting with programming a Bee-bot/Bluebot and learning how to give simple commands • Learning to debug instructions, with the help of an adult, when things go wrong	• Learning to debug an algorithm in an unplugged scenario	• Incorporating loops to make code more efficient	• Using a more systematic approach to debugging code, justifying what is wrong and how it can be corrected	• Incorporating variables to make code more efficient • Remixing existing code • 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	• Changing a program to personalise it • Evaluating code to understand its purpose • Predicting code and adapting it to a chosen purpose • Altering a website's code to create changes