



## Computing Progression Map

	Reception	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
<b>Information Technology</b>	<p>complete a simple activity/game on a digital device.</p> <p>use software to create digital content (art, music etc).</p> <p>shoot and review digital photos.</p> <p>type letters using a keyboard (physical or touchscreen) to write short words or sentences.</p>	<p>recognise that digital content is represented in many forms (image, text, audio, video).</p> <p>use a variety of software to manipulate and present digital content.</p> <p>talk about my work and make changes to improve it. I can name, save and find my work.</p>	<p>use a range of input and output devices (keyboard, mouse, touchscreen, microphone, screen, printout, video, audio etc).</p> <p>organise and find data using specific searches (e.g. using 2Investigate).</p> <p>use several programs to organise information (e.g. using binary trees such as 2Question or spreadsheets such as 2Calculate).</p>	<p>carry out searches to find digital content on a range of online systems (e.g. on an internet search engine).</p> <p>collect, organise and presents data and information in digital content.</p> <p>talk about my work and make improvements based on feedback received.</p> <p>present and analyse data and information using different software (e.g.</p>	<p>understand the purpose of a search engine and the main features within it.</p> <p>look at information on a webpage and make predictions about the accuracy of information contained within it.</p> <p>create and improve my solutions to a problem based on feedback (e.g. creating a program using Scratch or 2Code).</p>	<p>search precisely when using a search engine (e.g. I know I can add additional words or remove words to help find better results).</p> <p>explain in detail how accurate, safe and reliable the content is on a webpage.</p> <p>recognise the audience when designing and creating digital content.</p> <p>comment on how successful a digital solution is that I have created</p>	<p>use filters when searching for digital content and can use more complex searches for information (e.g. 'AND', 'OR', 'NOT').</p> <p>explain in detail how accurate and reliable a webpage and its content is.</p> <p>compare a range of digital content sources and rate them in terms of content quality and accuracy.</p> <p>consider the intended audience carefully when</p>



			edit digital data (e.g. data in music composition software like 2Sequence).	2Question - branching database or 2Graph - graphing tool).	review solutions that others have created, using a checklist of criteria	(e.g. a program built in 2Code or Scratch).	I design and make digital content.
<b>Computer Science (Theory)</b>	<p>identify the main parts of a computer system (monitor, mouse, keyboard, printer etc).</p> <p>recognise that a range of technology is used in places such as homes and schools.</p> <p>share my experiences of observing technology in school and</p>	<p>recognise and name common input and output devices of computer system.</p> <p>recognise the difference between old and new technology (e.g. typewriter and smartphone).</p> <p>explain how people interact with computers (e.g. cashpoint machine, self-service scanners etc).</p>	<p>explain the functions of the main components of a computer system.</p> <p>understand that my creations (e.g. programs), need similar skills to the adult world (e.g. the ParentPay etc used for collecting money for school trips).</p> <p>describe uses of technology beyond school.</p>	<p>understand the difference between hardware and software and their roles within computer system.</p> <p>use communication tools (such as 2Email) to attach files and use tools respectfully and with good etiquette.</p> <p>identify different ways that the</p>	<p>recognise the main component parts of hardware which allow computers to join and form a network.</p> <p>understand that network and communication components can be found in many different devices which allow them to join the internet.</p>	<p>know that computers collect data from various input devices.</p> <p>know the importance of computer networks and how they help solve problems and enhance communication .</p> <p>recognise the main dangers that can be perpetuated via computer networks.</p> <p>use the most</p>	<p>understand why and when computers are used and understand the main functions of the operating system.</p> <p>know the difference between physical, wireless and mobile networks.</p> <p>explain the difference between the</p>



	beyond the classroom.		(e.g. explain how a supermarket or airport uses ICT to help it operate). identify uses of technology beyond school (e.g. wearable technology, robots, drones, simulations etc) and discuss reasons why they are helpful . *** iWristband	internet can be used for communication confidently share my own experiences of technology in school and beyond the classroom.	understand how the Internet works, including how it is structured and how data travels along it. understand how search engines operate, including how they rank results.  show an awareness of tasks best completed by humans or computers.	appropriate form of online communication according to the digital content. (e.g. 2Email, 2Blog).  find out about the history of computing, including pioneers in developing different technologies.	internet and the World Wide Web.  explain what a WAN (Wide Area Network) and LAN (Local Area Network) is and describe the process of how access to the internet in school is possible.  describe the services offered by the Internet.
<b>Computer Science (Coding and Computational Thinking)</b>	follow given instructions to program a physical device.  understand what an algorithm is.  demonstrate an ability to	explain that an algorithm is a set of precise step-by-step instructions to achieve a particular task.  know that an algorithm written for a computer is	understand that algorithms are implemented on digital devices as programs and can identify examples of each.	make a real-life situation into an algorithm for a program.  design an algorithm carefully, thinking about what I want it to do and how	turn a real-life situation to solve into an algorithm, using a diagram to express solutions.  use repetition in my code. For example, using	make more complex real-life problems into algorithms for a program.  test and debug my programs as I work.  convert (translate)	turn a complex programming task into an algorithm.  identify the important aspects of a programming task (abstraction).



	<p>following an algorithm.</p> <p>design simple algorithms.</p> <p>detect and corrects errors (debugging) in simple algorithms.</p>	<p>called a program.</p> <p>work out what is wrong when the steps are out of order in instructions. say that if something does not work how it should, it is because my code is incorrect.</p> <p>try and fix my code if it isn't working properly (debugging).</p> <p>make good guesses (logical reasoning) of what is going to happen in a program. For example, where the Bee- Bot might go.</p>	<p>know I need to carefully plan my algorithm so it will work when I make it into code.</p> <p>design a simple program (e.g. using 2Code) that achieves a purpose.</p> <p>find and correct some errors in my program (debugging). I can say what will happen in a program.</p> <p>spot something in a program that has an action or effect (does something).</p>	<p>I can turn it into code.</p> <p>design a program thinking logically about the sequence of steps required. experiment with timers in my programs.</p> <p>experiment with the effect of using repeat commands.</p> <p>identify the difference in using the effect of a timer or repeat command in my code.</p> <p>identify an error in my program and fix it.</p> <p>read programs with several</p>	<p>a loop that continues until a condition is met such as the correct answer being entered.</p> <p>use timers within my program designs more accurately to create repetition effects.</p> <p>use selection (decision) in my programming. For example, using an 'if statement' for a question being asked and the program takes one of two paths.</p> <p>use variables within my program and</p>	<p>algorithms that contain sequence, selection and repetition into code that works.</p> <p>use sequence, selection, repetition, and some other coding structures in my code.</p> <p>organise my code carefully for example, naming variables and using tabs. I know this will help me debug more efficiently.</p> <p>use logical methods to identify the cause of any bug with support to</p>	<p>decompose important aspects of a programming task in a logical way, identifying appropriate coding structures that would work.</p> <p>test and debug my program as I work on it and use logical methods to identify a cause of a bug.</p> <p>identify a specific line of code that is causing a problem in my program and attempt a fix.</p> <p>translate algorithms that include sequence, selection and</p>
--	---	---	--	--	---	---	--



				<p>steps and predict what it will do.</p>	<p>know how to change the value of variables.</p> <p>use the user inputs and output features within my program, such as 'Print to screen'.</p> <p>identify errors in my code by using different methods, such as stepping through lines of code and fixing them.</p> <p>read programs that contain several steps and predict the outcomes with increasing accuracy.</p>	<p>identify the specific line of code.</p>	<p>repetition into code and nest these structures within each other.</p> <p>use inputs and outputs within my coded programs such as sound, movement and buttons and represent the state of an object.</p> <p>interpret (understand) a program in parts and can make logical attempts to put the separate parts together in an algorithm to explain the program as a whole.</p>
--	--	--	--	---	---	--	--