Old Moat Primary School

	Computing					
	Skills Progression					
	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Internet safety	The child can keep themselves safe while using digital technology. The child can understand that they need to keep safe when using digital technology. E.g. They should know to use filtered SafeSearch when looking for images on the web and that they should close the lid of a laptop (or similar action) if they find inappropriate images.	The child can keep safe and show respect to others while using digital technology. The child should know that they need to keep themselves safe when using digital technology. E.g. They should know to use filtered SafeSearch when looking for images on the web and that they should close the lid of a laptop (or similar action) if they find inappropriate images. They should know to respect others' rights, including privacy and intellectual property when using computers, so should not look at	The child can use digital technology safely and show respect for others when working online. The child should know that they need to keep themselves safe when using digital technology. E.g. They should show respect for others when filming and should not normally post videos online. They should take care when using the Command prompt and should treat links and attachments in emails with caution. If responding to online surveys, they should do so anonymously,	The child can demonstrate that they can act responsibly when using computers. The child can act responsibly when using computers. E.g. They should act responsibly when developing computer games or prototype products. They should behave responsibly when using sampled music or creating a composition. They should show responsibility when creating or remixing online content,	The child can demonstrate that they can act responsibly when using the internet. The child can act responsibly when using the internet. E.g. They should act responsibly when participating in an online community, such as the Scratch community, if permitted to do so. They should demonstrate that they understand the importance of encrypted (HTTPS) connections when browsing	The child can show that they can think through the consequences of their actions when using digital technology. The child can discuss likely and potential consequences of their actions when using digital technology in a range of contexts. Contexts might include developing smartphone apps; using online project management tools; collecting information for market research; posting original content online.

		someone else's work or copy it without permission and acknowledgement. They should observe age restrictions on computer games.	thinking carefully about information they give out.	including observing copyright and any terms and conditions. They should contribute positively to a shared wiki.	the web and of using strong passwords to protect their identity online. They should act responsibly when creating, editing or commenting on web pages or blog posts.	
Understanding algorithms	The child can understand algorithms as sequences of instructions in everyday contexts. The child can take real-world problems and then plan a sequence of steps to solve these. The problems could be moving a Bee Bot from one point to another, or making some simple food items like a sandwich, smoothie or pizza.	Understand what algorithms are. The child can understand how algorithms are implemented as programs on digital devices, and that programs execute by following precise and unambiguous instructions. Use logical reasoning to predict the behaviour of simple programs.	Use sequence, selection and repetition in programs; work with variables. Use logical reasoning to detect and correct errors in algorithms and programs. Understand computer networks including the internet. Understand how networks can provide multiple services, such as the world wide web.	The child can explain an algorithm using sequence and repetition in their own words. Given an algorithm using both sequence and repetition, the child can give a coherent, logically reasoned explanation of what it does and how it works. Repetition is likely to be 'forever' or for a set number of times, although end conditions (e.g.	The child can explain a rule-based algorithm in their own words. When provided with a rule-based algorithm (e.g. for a computer game), the child should be able to explain what it does and how it works, in their own words.	The child can give clear and precise logical explanations of a number of algorithms. Given an algorithm, the child can describe what it does and, using logical reasoning, give precise explanations of how it works. Algorithms could be linked to programming projects, but might include a key algorithm such as binary search.

	(E.g. In 1.1, recognise a set of directions as an algorithm. In 1.2, recognise the steps of a recipe as an algorithm. In 1.4, realise that there are algorithms for grouping or sorting things.)			repeat...until...) could be used.		
Create and debug programmes	An understanding of the word debugging	An understanding of the word debugging	Design, write and debug programs that accomplish specific goals. Solve problems by decomposing them into smaller parts	The child can design and write a program using a block language to a given brief, including simple interaction. The child can write a program in Scratch (or similar) in which the user has to provide some input, perhaps as an answer to a question on screen, or by using key presses or the	The child can design, write and debug a program using a block language based on their own ideas. The child can design a program of their own and write this in a block-based language such as Scratch. The child can test and debug their code, explain what bugs they found and how they fixed	The child can design, write and debug a program using a second programming language based on their own ideas. The child can design a program of their own and write this in a programming language other than Scratch (or whichever language has formed the focus for their programming in other years), such as TouchDevelop or App Inventor. The second language does not need to be text based, but Logo

				mouse. The program could be a simple game or a set of questions and typed responses.	them. The program need not be complex (a simple game or a turtle graphics program would suffice) but it should be accomplished with a degree of independent working.	or Python could be used. The child can test and debug their code, explain what bugs they found and how they fixed these. The program need not be complex - a simple app would suffice.
Logic and reasoning	The child can give explanations for what they think a program will do. The child can explain to the teacher, and to peers, what they think a program will do. This could be a program they or their peers have written, or it could be a familiar piece of software (including computer games). The child could	The child can give logical explanations for what they think a program will do. The child can give logical explanations of what a program will do under given circumstances, including some attempt at explaining why it does what it does. The program could be one they themselves have written or it could be a computer game or a familiar piece of software. The child could use an audio recorder or a video camera	The child can use logical reasoning to detect errors in programs. The child can give well-thought-through reasons for errors they find in programs. Typically, the child can find errors by reasoning logically about the program code, but they might also be able to use logical reasoning to identify errors in programs when they are executed. The programs do not have to be written originally by the child.	The child can use logical reasoning to detect and correct errors in programs. The child can give well-thought-through reasons for errors they find in programs and explain how they have fixed these. The child can find and correct errors by reasoning logically about the program code; they might also be able to use	The child can use logical reasoning to detect errors in algorithms. When given an algorithm for a particular purpose, e.g. a rule-based algorithm for a computer game or a sequence of steps to draw a geometric pattern, the child can use logical reasoning to identify possible errors in the algorithm, explaining why they believe	The child can use logical reasoning to detect and correct errors in algorithms (and programs). When given an algorithm for a particular purpose, e.g. a rule-based algorithm for a smartphone app, the child can use logical reasoning to identify possible errors in the algorithm, explaining why they believe the algorithm is incorrect. The child can use logical reasoning to suggest possible corrections to the algorithm, explaining why these

	use an audio recorder or video camera to capture their explanations.	to record their explanations.		logical reasoning to identify errors in programs when executed and confirm that they have fixed these by testing the new version of their program. The programs do not have to be written originally by the child.	the algorithm is incorrect.	would correct the bug they identified.
Designing programmes	The child can give a sequence of instructions to a floor turtle. The child can create a Bee Bot program using a sequence of instructions before running it using the Go button. The length of the child's programs might be expected to increase over the course of the year.	The child can create a simple program on screen, correcting any errors. The child can create a simple program on screen (e.g. using the Blue Bot app, ScratchJr or with prepared sprites and blocks in Scratch) with a particular goal or purpose in mind (e.g. drawing a shape or moving a sprite from one place to another). The child can debug any errors in their own code.	The child can use sequence in programs. In on-screen programming, the child's program should include a sequence of commands or blocks in an appropriate order. A typical program could be a simple scripted animation, e.g. telling a joke, a story or explaining an idea taken from elsewhere on the curriculum. The child's program might include multiple sprites;	The child can use sequence and repetition in programs. The child's program, typically written in Scratch, or similar, should include sequences of commands or blocks and some repetition. Repetition would typically be for a fixed number of times, but might also include exit conditions (e.g.	The child can use sequence, selection and repetition in programs. The child's program, typically written in Scratch, or similar, should include sequences of commands or blocks, some repetition and selection. Repetition might include exit conditions (e.g. repeat...until...). Selection would normally	The child can use sequence, selection, repetition and variables in programs. The child's program should include sequences of commands or blocks, repetition, selection and variables. Repetition might include exit conditions (e.g. repeat...until...) and perhaps a counter-variable for iteration. Selection would normally be of an if...then or if...then...else type. At this level, expect the child to be able

			instructions could include movement, on-screen text, sound and/or costume changes.	repeat...until...). Programs might include turtle graphics, simple music or a simple game.	be of an if...then or if...then...else type. At this level, expect the child to be able to combine repetition with selection. Programs might include a computer game or a turtle graphics design.	to combine repetition with selection and variables. Programs might include a simple smartphone app.
Understanding search engines	To complete a simple search using a range of search engines.	To complete a simple search using a range of search engines.	The child can search for information within a single site. The child can use browser-specific tools (e.g. the Find command) and site-specific tools (such as the search tools for Wikipedia or YouTube) to locate particular information on a web page or within a website.	The child can use a standard search engine to find information. The child can use a common search engine (such as Google with safe search mode locked in place) effectively, to search for particular information on the web, such as answers to questions they identify in a research project.	The child can use filters to make more effective use of a standard search engine. The child can use a common search engine (such as Google with safe search mode locked in place) effectively, to search for particular information on the web, such as answers to questions they identify in a research	The child can make use of a range of search engines appropriate to finding information that is required. The child can show that they can use effectively a range of different search technologies, including alternatives to Google (such as Bing or Yahoo) and site-specific search engines (such as those for the App Store or Google Play). E.g. They could demonstrate how they would use a range of search

					project. They should use built-in search tools to filter their results, such as by time, location or reading level.	engines when researching available smartphone apps for a particular purpose.
Understanding computer networks	The child can use digital technology to store and retrieve content. The child can use a range of digital technologies to store and access digital content. These might include laptop computers, tablets, smartphones, digital cameras, video cameras and audio recorders. Projects might include videoing one another cooking, developing an	The child can store, organise and retrieve content on digital devices for a given purpose. With a given purpose, the child can use a range of digital technologies to retrieve, organise and store digital content. Technologies will typically include laptop computers, tablets and smartphones with access to the internet, but the child might also be expected to use digital cameras, video cameras and audio recorders (or the equivalent apps on a tablet or smartphone). Projects might include digital photography,	The child can use a range of programs on a computer. The child can use a range of software on laptop or tablet computers with some degree of independence. Software might include video editing, diagnostic tools, email clients, videoconferencing (with the teacher or another adult), survey design software, spreadsheets and presentation software. The child can understand that computer networks transmit	The child can use a range of programs on a computer. The child can use a range of software on laptop or tablet computers, possibly with some support as appropriate. Software might include audio editing, music composition, web browsers, text editors, spreadsheets and presentation software. he child can understand that the internet transmits information as	The child can use and combine a range of programs on multiple devices. The child can use multiple digital devices (such as tablets and laptops or digital cameras and laptops) to achieve particular goals. The devices might include web servers, allowing them to use cloud-based applications. E.g. They might use local media in conjunction with a cloud-based	The child can use and combine a range of programs on multiple devices. The child can use multiple digital devices (such as tablets and laptops or digital cameras and laptops) to achieve particular goals. The devices might include web servers, allowing them to use cloud-based applications. E.g. They might use local media to make a presentation using cloud-based presentation software, such as Google Slides, local media, cloud-based programming environments and a connected tablet or smartphone to help in developing and testing an app; a

	e-book or an audio book, creating a greetings card.	searching for images online and creating image-based presentation slides.	information in a digital (binary) format. The child can explain that any information has to be converted to numbers before it can travel through computer networks. The child should understand that this conversion happens according to an agreed system or code.	packets of data. When working online, the child can explain that the information they send and receive is automatically broken down into packets of data, and that these sometimes take different routes across the internet.	programming platform, such as Scratch; digital cameras and video cameras to capture content to use on an externally hosted website or blog; a digital camera to take photos they could import into 3D design software on a laptop. The child can understand how data routing works on the internet. The child can give a coherent explanation of how data packets are routed from one computer to another on a separate network, which is also connected to the internet.	video camera, laptop-based editing software and online video streaming to create a marketing video for an app. The child can understand how mobile phone or other networks operate. The child can give an explanation of how mobile phone (or other) networks operate: they should know that information is transmitted digitally, and have some understanding of the network topology involved. In the case of mobile phone networks, the child should show some understanding of the interactions between a phone, cell transmitters/receivers and the network's control systems.