WHITLEY MEMORIAL C of E AIDED PRIMARY SCHOOL

'Let your light shine' – Matthew 5:16

COMPUTING- Computer Science

Overview	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.

Year Group	YEAR 1	YEAR 2	YEAR 3	YEAR 4	YEAR 5	YEAR 6
Key Learning: National Curriculum knowledge covered **Key Progressive Skills:** National Curriculum skills covered	**Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions.** *(Children understand that an algorithm is a set of instructions used to solve a problem or achieve an*	**Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions.** *Children can explain that an algorithm is a set of instructions to complete a task. When* **designing**	**Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.** *Children can turn a simple real-life situation into an algorithm for a program by deconstructing it into manageable parts. Their*	**Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.** When turning a reallife situation into an algorithm, the children's design shows that they are thinking of the	**Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.** Children may attempt to turn more complex real-life situations into algorithms for a program by deconstructing it into	**Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.** Children are able to turn a more complex programming task into an algorithm by identifying the important aspects of the task (abstraction) and then decomposing them in a logical way using their knowledge of possible coding structures and applying skills from previous programs. Children test and debug their

	objective. They know that an algorithm written for a computer is called a program.) **Create and debug simple programs.** *Children can work out what is wrong with a simple algorithm when the steps are out of order, e.g. The Wrong Sandwich in Purple Mash and can write their own simple algorithm* **Use logical reasoning to predict the behaviour of simple programs.** *When looking at a program, children can read code one line at a time and make good attempts to envision the bigger picture of the overall effect of the program. Children can, for example, interpret where the turtle in 2Go challenges will end up at the end of the program.*	**simple programs**, *children show an awareness of the need to be precise with their algorithms so that they can be successfully converted into code.* **Create and debug simple programs.** *Children can create a simple program that achieves a specific purpose. They can also identify and correct some errors, e.g.* **Debug Challenges: Chimp**. *Children's program designs display a growing awareness of the need for logical, programmable steps.* **Use logical reasoning to predict the behaviour of simple programs.** *Children can identify the parts of a program that respond to specific events and initiate specific actions. For example, they can write a cause and effect sentence of what will happen in a program.*	*design shows that they are thinking of the desired task and how this translates into code. Children can identify an error within their program that prevents it following the desired algorithm and then fix it.* **Use sequence, selection and repetition in programs; work with variables and various forms of input and output.** *Children demonstrate the ability to design and code a program that follows a simple sequence. They experiment with timers to achieve repetition effects in their programs. Children are beginning to understand the difference in the effect of using a timer command rather than a repeat command when creating repetition effects. Children understand how variables can be used to store information while a program is executing.* **Use logical reasoning to explain how some simple algorithms work and to detect and correct**	required task and how to accomplish this in code using coding structures for selection and repetition. Children make more intuitive attempts to debug their own programs. **Use sequence, selection and repetition in programs; work with variables and various forms of input and output.** Children's use of timers to achieve repetition effects are becoming more logical and are integrated into their program designs. They understand 'if statements' for selection and attempt to combine these with other coding structures including variables to achieve the effects that they design in their programs. As well as understanding how variables can be used to store information while a program is executing, they are able to use and manipulate the value of variables. Children can make use of user inputs and outputs such as 'print to screen'. e.g. 2Code. **Use logical reasoning to**	manageable parts. Children are able to test and debug their programs as they go and can use logical methods to identify the approximate cause of any bug but may need some support identifying the specific line of code. **Use sequence, selection and repetition in programs; work with variables and various forms of input and output.** Children can translate algorithms that include sequence, selection and repetition into code with increasing ease and their own designs show that they are thinking of how to accomplish the set task in code utilising such structures. They are combining sequence, selection and repetition with other coding structures to achieve their algorithm design. **Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.**	program as they go and use logical methods to identify the cause of bugs, demonstrating a systematic approach to try to identify a particular line of code causing a problem. **Use sequence, selection and repetition in programs; work with variables and various forms of input and output.** Children translate algorithms that include sequence, selection and repetition into code and their own designs show that they are thinking of how to accomplish the set task in code utilising such structures, including nesting structures within each other. Coding displays an improving understanding of variables in coding, outputs such as sound and movement, inputs from the user of the program such as button clicks and the value of functions. **Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.** Children are able to interpret a program in parts and can make logical attempts to put the separate parts of a complex algorithm together to explain the program as a whole. **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.** Children understand and can explain in some depth the difference between the internet and the World Wide Web. Children know what a WAN and LAN are and can describe how they access the internet in school.

			errors in algorithms and programs. *Children's designs for their programs show that they are thinking of the structure of a program in logical, achievable steps and absorbing some new knowledge of coding structures. For example, 'if' statements, repetition and variables. They make good attempts to 'step through' more complex code in order to identify errors in algoriams and can correct this. e.g. traffic light algorithm in* **2Code.** *In programs such as Logo, they can 'read' programs with several steps and predict the outcome accurately.* **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.** *Children can list a range of ways that the internet can be used to provide different methods of communication. They can*	**explain how some simple algorithms work and to detect and correct errors in algorithms and programs.** Children's designs for their programs show that they are thinking of the structure of a program in logical, achievable steps and absorbing some new knowledge of coding structures. For example, 'if' statements, repetition and variables. They can trace code and use step-through methods to identify errors in code and make logical attempts to correct this. e.g. traffic light algorithm in 2Code. In programs such as Logo, they can 'read' programs with several steps and predict the outcome accurately. **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.** Children recognise the main component parts of	When children code, they are beginning to think about their code structure in terms of the ability to debug and interpret the code later, e.g. the use of tabs to organise code and the naming of variables **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.** Children understand the value of computer networks but are also aware of the main dangers. They recognise what personal information is and can explain how this can be kept safe. Children can select the most appropriate form of online communications contingent on audience and digital content, e.g. 2Blog, 2Email, Display Boards.	

			use some of these methods of communication, e.g. being able to open, respond to and attach files to emails in **2Email**. *They can describe appropriate email conventions when communicating in this way.*	hardware which allow computers to join and form a network. Their ability to understand the online safety implications associated with the ways the internet can be used to provide different methods of communication is improving.		
Programs/ equipment used	**Daisy Dinosaur** **Beebots** **Ipads – Beebot Program** **Scratch 2.0 (make your own simulator)** **J2 Code**		**Scratch.** **Networks glossary.** **Networks sorting activity.**			
Key Vocabulary National Curriculum and other	**Debug** **Program** **Programmable** **Toy** **Forwards** **Backwards** **Right Left** **Algorithm – step by step instructions.** **Fix** **Input** **Record** **sequence**		**Sprites, blocks, loops, code, debug, programming, replicate. Network, wireless, components, files, device**			

WHITLEY MEMORIAL C of E AIDED PRIMARY SCHOOL

‘Let your light shine’ – Matthew 5:16