

Computing Progression Sample

		EYFS	KS1	LKS2	UKS2
Computer Science	(A) Algorithms (Making steps and rules)		Children understand that an algorithm is a set of instructions used to solve a problem or achieve an objective. They know that an algorithm written for a computer is called a program .	When turning a real-life situation into an algorithm , the children's design shows that they are thinking of the 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 .	Children may attempt to turn more complex real-life situations into algorithms for a program by deconstructing it into 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 .
	(B) Debugging (Fixing and finding errors)	<ul style="list-style-type: none"> • Completes a simple program on a computer. • Uses ICT hardware to interact with age-appropriate computer software. Children recognise that a range of technology is used in places such as homes and schools. They select and use technology for particular purposes. 	Children can work out what is wrong with a simple algorithm when the steps are out of order and can write their own simple algorithm. Children know that an unexpected outcome is due to the code they have created and can make logical attempts to fix the code.	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'.	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.
	(C) Logic (Predicting and analysing)		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'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. They can 'read' programs with several steps and predict the outcome accurately.	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 .