

	Computer Science				Information Technology	Digital Literacy	
Statement	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 concern about content and contact.
Possible Outcomes	Children can turn a simple real-life situation into an algorithm for a program by deconstructing it into manageable parts. Their 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.	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.	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 algorithms and can correct this. e.g. traffic light algorithm in <a href="#">2Code</a> . In programs such as Logo, they can 'read' programs with several steps and predict the outcome accurately.	Children can list a range of ways that the internet can be used to provide different methods of communication. They can use some of these methods of communication, e.g. being able to open, respond to and attach files to emails using <a href="#">2Email</a> . They can describe appropriate email conventions when communicating in this way.	Children can carry out simple searches to retrieve digital content. They understand that to do this, they are connecting to the internet and using a search engine such as Purple Mash search or internet-wide search engines.	Children can collect, analyse, evaluate and present data and information using a selection of software, e.g. using a branching database ( <a href="#">2Question</a> ), using software such as <a href="#">2Graph</a> . Children can consider what software is most appropriate for a given task. They can create purposeful content to attach to emails, e.g. <a href="#">2Respond</a> .	Children demonstrate the importance of having a secure password and not sharing this with anyone else. Furthermore, children can explain negative implications of failure to keep passwords safe and secure. They understand the importance of staying safe and the importance of their conduct when using familiar communication tools such as <a href="#">2Email</a> in Purple Mash. They know more than one way to report unacceptable content and contact.

