



Computing Progression & Skills Document



	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Computer Science	<ul style="list-style-type: none"> Understand that an algorithm is a set of instructions used to solve a problem or achieve an objective. Know that a computer program turns an algorithm into code that the computer can understand Work out what is wrong with a simple algorithm when the steps are out of order Know that an unexpected outcome is due to the code they have created and can make logical attempts to fix the code Read code one line at a time and make good attempts to envision the bigger picture of the overall effect of the program. 	<ul style="list-style-type: none"> Explain that an algorithm is a set of instructions to complete a task. Show an awareness of the need to be precise with their algorithms so that they can be successfully converted into code. Create a simple program that achieves a specific purpose. Identify and correct some errors Display a growing awareness of the need for logical, programmable steps. Identify the parts of a program that respond to specific events and initiate specific actions. 	<ul style="list-style-type: none"> Turn a simple real-life situation into an algorithm by deconstructing it into manageable parts. Identify an error within their program that prevents it following the desired algorithm and then fix it. Demonstrate the ability to design and code a program that follows a simple sequence. Experiment with timers to achieve repetition effects in their programs. Design programs by thinking of the structure of a program in logical, achievable steps and absorbing some new knowledge of coding structures. List a range of ways that the Internet can be used to provide different methods of communication. Use some methods of communication, e.g. 2Email. 	<ul style="list-style-type: none"> Design an algorithm thinking of the required task and how to accomplish this in code using coding structures for selection and repetition. Make intuitive attempts to debug their own programs. Use timers to achieve repetition effects and integrate into program designs. 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. 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. Make use of user inputs and outputs such as 'print to screen'. e.g. 2Code. Recognise the main component parts of hardware which 	<ul style="list-style-type: none"> Begin to turn more complex real-life situations into algorithms for a program by deconstructing it into manageable parts. Test and debug their programs as they go and use logical methods to identify the approximate cause of any bug with some support. Translate algorithms that include sequence, selection and repetition into code with increasing ease. Combine sequence, selection and repetition with other coding structures to achieve their algorithm design. Consider 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 the value of computer networks with awareness of the main dangers. 	<ul style="list-style-type: none"> 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. Test and debug their 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. 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. Interpret a program in parts and can make logical

				<p>allow computers to join and form a network.</p> <ul style="list-style-type: none"> • Understand the online safety implications associated with the ways the internet can be used to provide different methods of communication 	<ul style="list-style-type: none"> • Recognise what personal information is and can explain how this can be kept safe. • Select the most appropriate form of online communication e.g. 2Blog, 2Email, Display Boards. 	<p>attempts to put the separate parts of a complex algorithm together to explain the program as a whole.</p> <ul style="list-style-type: none"> • Explain the difference between the internet and the World Wide Web. • Know what a WAN and LAN are and can describe how they access the Internet in school
<ul style="list-style-type: none"> • Sort, collate, edit and store simple digital content • Name, save and retrieve their work and follow simple instructions to access online resources, • Use Purple Mash 2Quiz, 2Code, 2Count. 	<ul style="list-style-type: none"> • Demonstrate an ability to organise data using, e.g. 2Investigate • Retrieve specific data for conducting simple searches. • Edit complex digital data, e.g. 2Sequence • Create, name, save and retrieve content. • Use a range of media in their digital content including photos, text and sound. 	<ul style="list-style-type: none"> • Carry out simple searches to retrieve digital content, using a search engine. • Collect, analyse, evaluate and present data and information using a selection of software, e.g. 2Question, 2Graph. • Consider what software is most appropriate for a given task. • Create purposeful content to attach to emails, e.g. 2Respond. 	<ul style="list-style-type: none"> • Understand the function, features and layout of a search engine. • Appraise selected webpages for credibility and information at a basic level. • Make improvements to digital solutions based on feedback. • Make informed software choices when presenting information and data. • Create linked content using a range of software, e.g. 2Connect, 2Publish+. • Share digital content within their community, i.e. using Virtual Display Boards. 	<ul style="list-style-type: none"> • Search with greater complexity for digital content when using a search engine. • Explain in some detail how credible a webpage is and the information it contains. • Make appropriate improvements to digital solutions based on feedback received and comment on the success of the solution. E.g. 2Code. • Review solutions from others. • Collaboratively create content and solutions using digital features within software such as collaborative mode. • Use several ways of sharing digital content, i.e. 2Blog, 2Email. 	<ul style="list-style-type: none"> • Apply filters when searching for digital content. • Explain in detail how credible a webpage is and the information it contains. • Compare a range of digital content sources and rate them in terms of content quality and accuracy. • Use critical thinking skills in everyday use of online communication. • Make clear connections to the audience when designing and creating digital content. • Design and create their own blogs to become a content creator on the Internet, e.g. 2Blog. • Use criteria to evaluate the quality of digital solutions and are able to identify improvements, 	

						making some refinements.
Digital Literacy	<ul style="list-style-type: none"> Understand what is meant by technology and can identify a variety of examples both in and out of school. Distinguish between objects that use modern technology and those that do not e.g. a microwave vs. a chair. Understand the importance of keeping information, such as their usernames and passwords, private and actively demonstrate this in lessons. Take ownership of their work and save this in their own private space. 	<ul style="list-style-type: none"> Retrieve relevant, purposeful digital content using a search engine. Share this knowledge, e.g. 2Publish Make links between technology they see around them, coding and multimedia work they do in school Know the implications of inappropriate online searches. Begin to understand how things are shared electronically such. Develop understanding of using email safely by using 2Respond Know ways of reporting inappropriate behaviours and content to a trusted adult 	<ul style="list-style-type: none"> Demonstrate the importance of having a secure password and not sharing this with anyone else. Explain the negative implications of failure to keep passwords safe and secure. Understand the importance of staying safe and the importance of their conduct when using familiar communication tools Know more than one way to report unacceptable content and contact. 	<ul style="list-style-type: none"> Explore key concepts relating to online safety using concept mapping, e.g. 2Connect. Help others to understand the importance of online safety. Know a range of ways of reporting inappropriate content and contact. 	<ul style="list-style-type: none"> Have a secure knowledge of common online safety rules and can apply this by demonstrating the safe and respectful use of a few different technologies and online services. Relate appropriate online behaviour to their right to personal privacy and mental wellbeing of themselves and others. 	<ul style="list-style-type: none"> Demonstrate the safe and respectful use of a range of different technologies and online services. Identify more discreet inappropriate behaviours through developing critical thinking, e.g. 2Respond. Recognise the value in preserving their privacy when online for their own and other people's safety.