

	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 Write a simple algorithm Know that an unexpected outcome is due to the code created and make logical attempts to fix the code When looking at a program, 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. When designing simple programs, show an awareness of the need to be precise with algorithms so that they can be successfully converted into code. Create a simple program that achieves a specific purpose. Identify and correct some errors. Program designs 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"> List a range of ways that the Internet can be used to provide different methods of communication. Use some of these methods of communication, e.g. open, respond to and attach files to emails Describe appropriate email conventions when communicating in this way. 	<ul style="list-style-type: none"> When turning a real-life situation into an algorithm, designs show thinking of the required task and how to accomplish this in code using coding structures for selection and repetition. Make intuitive attempts to debug own programs. Use of timers to achieve repetition effects are becoming more logical and are integrated into their program designs. Understand 'IF statements' for selection and attempt to combine these with other coding structures including variables to achieve the effects that are designed into programs. Understand how variables can be used to store information while a program is executing, use and manipulate the value of variables. Designs for programs show thinking of the structure of a program in logical, achievable steps and some new knowledge of coding structures is absorbed. Trace code and use step-through methods to identify errors in code and make logical attempts to correct this. Predict outcomes accurately Recognise the main component parts of hardware which allow computers to join and form a network. 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"> Attempt to turn more complex real-life situations into algorithms for a program by deconstructing it into manageable parts. Test and debug programs and use logical methods to identify the approximate cause of any bug but may need some support identifying the specific line of code. Translate algorithms that include sequence, selection and repetition into code with increasing ease and own designs show thoughts of how to accomplish the set task in code utilising such structures. Combine sequence, selection and repetition with other coding structures to achieve algorithm design. When coding, begin to think about 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 and aware of the main dangers. Recognise what personal information is and explain how this can be kept safe. Select the most appropriate form of online communications contingent on audience and digital content. 	<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 decompose them in a logical way using knowledge of possible coding structures and applying skills from previous programs. Test and debug programs 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 own designs show thought 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. Interpret a program in parts and make logical attempts to put the separate parts of a complex algorithm together to explain the program as a whole. Understand and explain in some depth the difference between the internet and the World Wide Web. Know what a WAN and LAN are and describe how the internet can be accessed in school.
Information Technology	<ul style="list-style-type: none"> Sort, collate, edit and store simple digital content e.g. a name, save and retrieve work and follow simple instructions to access online resources. 	<ul style="list-style-type: none"> Demonstrate an ability to organise data using, for example, a database and retrieve specific data for conducting simple searches. Edit more complex digital data such as music compositions. Confidence when creating, naming, saving and retrieving content. Use a range of media in digital content including photos, text and sound. 	<ul style="list-style-type: none"> Carry out simple searches to retrieve digital content. Collect, analyse, evaluate and present data and information using a selection of software, e.g. using a branching database. Consider what software is most appropriate for a given task. Create purposeful content to attach to emails. 	<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. 	<ul style="list-style-type: none"> Make appropriate improvements to digital solutions based on feedback received and confidently comment on the success of the solution. Objectively 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. 	<ul style="list-style-type: none"> Readily 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 blogs to become a content creator on the internet. Use criteria to evaluate the quality of digital solutions and identify improvements, making some refinements.
Digital Literacy	<ul style="list-style-type: none"> Understand what is meant by technology and identify a variety of examples both in and out of school. Make a distinction 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 usernames and passwords, private and actively demonstrate this in lessons. Take ownership of work and save this in private spaces. 	<ul style="list-style-type: none"> Effectively retrieve relevant, purposeful digital content using a search engine. Apply learning of effective searching beyond the classroom. Make links between technology, coding and multimedia work. Know the implications of inappropriate online searches. Begin to understand how things are shared electronically. Develop an understanding of using email safely and 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 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. 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 apply this by demonstrating the safe and respectful use of a few different technologies and online services. Implicitly relate appropriate online behaviour to rights to personal privacy and mental wellbeing. 	<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. Recognise the value in preserving privacy when online for people's safety.