Progression of Skills and Vocabulary for Computing

Key Stage	Computer Science statements	Information Technology Statements	Digital Literacy Statements
KS1	1. Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions. 2. Create and debug simple programs. 3. Use logical reasoning to predict the behaviour of simple programs.	1. Use technology purposefully to create, organise, store, manipulate and retrieve digital content.	1. Recognise common uses of information technology beyond school. 2. 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.
KS2	1. Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts. 2. Use sequence, selection and repetition in programs; work with variables and various forms of input and output. 3. Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs. 4. 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.	1. Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content. 2. 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.	1. Use technology safely, respectfully, and responsibly. recognise acceptable/ unacceptable behaviour; identify a range of ways to report concern about content and contact.

	Early Years	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Computer Science		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. 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. 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 explain that an algorithm is a set of instructions to complete a task. When designing simple programs, children show an awareness of the need to be precise with their algorithms so that they can be successfully converted into code. Children can create a simple program that achieves a specific purpose. They can also identify and correct some errors. Children's program design displays a growing awareness of the need for logical, programmable steps. 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.	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 is 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.	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'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'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.	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. 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. When children code, they are beginning to think about their code structure in terms of the ability to debug and interpret the code later. 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 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 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. 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 and 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. 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.

Computer Science				They make good attempts to 'step through' more complex code in order to identify errors in algorithms and can correct this. 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. They can describe appropriate email conventions when communicating in this way.	They can trace code and use step-through methods to identify errors in code and make logical attempts to correct this. Children recognise the main component parts of 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.	Children can select the most appropriate form of online communications contingent on audience and digital content.	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.
Information Technology		Children are able to sort, collate, edit and store simple digital content e.g. children can name, save and retrieve their work and follow simple instructions to access online resources.	Children demonstrate an ability to organise data using, for example, a database and can retrieve specific data for conducting simple searches. Children are able to edit more complex digital data such as music compositions. Children are confident when creating, naming, saving and retrieving content. Children use a range of media in their digital content including photos, text and sound.	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 internet-wide search engines. Children can collect, analyse, evaluate and present data and information using a selection of software. Children can consider what software is most appropriate for a given task. They can create purposeful content to attach to emails.	Children understand the function, features and layout of a search engine. They can appraise selected webpages for credibility and information at a basic level. Children are able to make improvements to digital solutions based on feedback. Children make informed software choices when presenting information and data. They create linked content using a range of software. Children share digital content within their community.	Children search with greater complexity for digital content when using a search engine. They are able to explain in some detail how credible a webpage is and the information it contains. Children are able to make appropriate improvements to digital solutions based on feedback received and can confidently comment on the success of the solution. e.g. creating their own program to meet a design brief. They objectively review solutions from others. Children are able to collaboratively create content and solutions using digital features within software such as collaborative mode. They are able to use several ways of sharing digital content.	Children readily apply filters when searching for digital content. They are able to explain in detail how credible a webpage is and the information it contains. They compare a range of digital content sources and are able to rate them in terms of content quality and accuracy. Children use critical thinking skills in everyday use of online communication. Children make clear connections to the audience when designing and creating digital content. The children design and create their own blogs to become a content creator on the internet, e.g. 2Blog. They are able to use criteria to evaluate the quality of digital solutions and are able to identify improvements, making some refinements.

Digital Literacy		Children understand what is meant by technology and can identify a variety of examples both in and out of school. They can make a distinction between objects that use modern technology and those that do not e.g. a microwave vs. a chair. Children understand the importance of keeping information, such as their usernames and passwords, private and actively demonstrate this in lessons. Children take ownership of their work and save this in their own private space.	Children can effectively retrieve relevant, purposeful digital content using a search engine. They can apply their learning of effective searching beyond the classroom. They can share this knowledge. Children make links between technology they see around them, coding and multimedia work they do in school. Children know the implications of inappropriate online searches. Children begin to understand how things are shared electronically. They develop an understanding of using email safely and know ways of reporting inappropriate behaviours and content to a trusted adult.	Children demonstrate the importance of having a secure password and not sharing this with anyone else. Furthermore, children can explain the 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. They know more than one way to report unacceptable content and contact.	Children can explore key concepts relating to online safety using concept mapping. They can help others to understand the importance of online safety. Children know a range of ways of reporting inappropriate content and contact	Children 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. Children implicitly relate appropriate online behaviour to their right to personal privacy and mental wellbeing of themselves and others.	Children demonstrate the safe and respectful use of a range of different technologies and online services. They identify more discreet inappropriate behaviours through developing critical thinking, e.g. 2Respond activities. They recognise the value in preserving their privacy when online for their own and other people's safety.

	Early Years	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
	*For definitions, please see knowledge organisers						
Vocabulary		action alert algorithm animation avatar background button calculations cell challenge clip-art code collect data column command compare computer criteria data debugging device direction E-book edit execute file name font groups icon instructions image lock cell log in log out menu my work area notification output password pictograms plan private program route row run	action algorithm attachment background binary tree block graph bug button cell collision detection column command copy count tool data database debug digital footprint domain drag E-book email equal event execute field filter implement instructions interaction internet interval label network node object output personal information pictogram presentation	action advance mode alert algorithm analysis attachment axis background bar graph BCC binary tree blog branching database bug button CC cell address chart code collision detection event command compose data database debug decision evaluation event flowchart font formatting implement inbox interval keys layer modelling nesting password permission pie chart posture predict presentation properties reputable source run	action alert algorithm attachment average background budget button campaign citation code blocks collaborate command components cookies copyright decimal place design digital footprint execute flowchart font format format cell formula formular wizard frame genre hard drive implement input internet key words malware motherboard multi-line mode nest network card onion skinning opinion output pause percentage peripherals phishing plagiarism	abstraction action algorithm avatar bulleted lists captions citation collaborate concatenation concept connection copy and paste copyright creative commons license cursor debug decomposition design brief efficient encrypt evaluation event field flowchart font format formatting formular wizard function hyperlink identify theft input instructions malware merge cells nesting net node output ownership page orientation password PEGI rating personal information phishing physical system	action algorithm approval archive audience audio auto fit bit blog blog post budget case-sensitive cell cell reference clone cloze co-ordinates collaborate command commenting computational model conditional formatting data analysis debug decomposition delimiter dice tool digit digital footprint execute expense flowchart format cell formular wizard function horizontal axis hub/switch inappropriate input integer launch command local area network (LAN) location sharing network output password

Vocabulary		sort sound effect spreadsheet technology text title undo unit value	private information properties question record row run search search engine secure sharing sort table total web address web page web site world wide web	save to draft scene sequence simulation slideshow space bar spoof spreadsheet transition trusted contact typing verify vlog website word art	predict procedure prompt RAM ransomware reliability results page run search engine selection sequence software spam variable viewpoint virus watermark	properties quest readability reliable source scene screenshot sequence simplify spoof statistics template text wrapping texture theme validity variable word processing tool	phishing predict preview print screen probability procedure profit properties range repeat router screen time secure website selection sequence simulation spoof sprite tab text-based adventure transistor variable vertical axis vlog wi-fi wide area network (WAN) world wide web