



Maybury Primary School

Computing Curriculum

The Computing Curriculum.

Our aim is for children to be enthusiastic, skilled, safe and socially responsible users of digital technology. We believe that they should be able to apply their knowledge to, and in turn, change the rapidly evolving digital world. We aim to ensure that children become digitally literate at a level suitable for the future workplace and as active participants in a digital world.

Our curriculum is designed so that children become more expert as they progress through the curriculum. As they do this, they accumulate and connect both substantive and disciplinary computing knowledge.

Substantive knowledge: This is subject knowledge and explicit vocabulary used to learn about the content

Disciplinary knowledge: This considers how computing knowledge originates and is revised. Through disciplinary knowledge, children gradually become more expert in their computational thinking and as creative, safe users of IT

In our school, substantive knowledge is developed through the key aspects of:

- **Computer Science**
- **Information Technology**
- **Digital Literacy**

Disciplinary knowledge is taught through a progression of identified **Key Skills** which underpin and support learning across the areas of the key substantive aspects.

Our curriculum is ambitious for all and strives to address inclusion and disadvantage in its intent and implementation. The curriculum at our school is planned, organised and taught in ways which are compatible with the Equality Act 2010 and the school's Equal Opportunities Policy. As a school, we take reasonable and necessary steps to ensure that all children can access a broad and balanced curriculum. This includes ensuring that the environment is accessible as well as lesson content. In some cases we may consult with our SENCo or external agencies for advice to meet the needs of some children to ensure that they are able to participate in all lessons across the curriculum. A wide variety of strategies are used to ensure that teaching meets the needs of different groups of children including those that are identified with special educational needs. These include:

- Using a range of resources appropriate to need
- Ensuring that tasks set are accessible to all and include scaffolding as appropriate
- 'Pre-loading' with identified children
- Using a range of groupings within the class to teach children and support them
- Targeted adult support which is adapted as necessary

Our computing curriculum is based upon the scheme of learning devised by Purple Mash and is supplemented by resources from National Online Safety and Teach Computing.

Children do not necessarily have a timetabled computing lesson each week. In some weeks, computing can be covered by using technology to demonstrate learning in other subjects.



Maybury Primary School

Long Term Plan: Teaching Unit Plan Overview



EYFS Long Term Plan: Teaching Unit Plan Overview

	Computer Science	Information Technology	Digital Literacy
	<ul style="list-style-type: none">• Help adults operate equipment around the school, independently operating simple equipment• Use simple software to make things happen• Press buttons on a floor robot and talk about the movements• Explore options and make choices with toys, software and websites	<ul style="list-style-type: none">• Use a mouse to rearrange objects and pictures on screen• Use a camera or sound recorder to collect photos or sound• Use paint programs to create pictures• Begin to use a keyboard	<ul style="list-style-type: none">• Talk about good and bad choices in life e.g. taking turns, saying kind things, helping others, telling an adult if something upsets you• Play appropriate games on the internet• Talk about good and bad choices when using websites – being kind, telling an adult if something upsets us and keeping ourselves safe by keeping information private



Long Term Plan: Teaching Unit Plan Overview

	Autumn			Spring		Summer		
Y1	1.1 Online Safety	1.4 Lego Builders		1.8 Spreadsheets		1.7 Coding		
Y2	2.2 Online Safety	2.1 Coding		2.5 Effective searching		2.3 Spreadsheets	2.7 Making Music	
Y3	3.2 Online Safety	3.1 Coding		3.4 Typing	3.5 Email	3.3 Spreadsheets	3.6 Branching Databases	
Y4	4.2 Online Safety	4.1 Coding	4.8 Hardware Investigators	4.5 Logo	4.8 Hardware Investigators	4.3 Spreadsheets	4.9 Making Music	4.8 Hardware Investigators
Y5	5.2 Online Safety		5.1 Coding	5.5 Game Creator	5.8 Word Processing	5.3 Spreadsheets		
Y6	6.2 Online Safety		6.1 Coding	6.6 Networks	6.8 Binary (link to 6.7 – Quizzing)	6.3 Spreadsheets	6.4 Blogging	

Computer Science	
Information Technology	
Digital Literacy	



Maybury Primary School

Progression in Computing Knowledge and Skills

	Key Stage 1	Key Stage 2
Computer Science	<ul style="list-style-type: none"> • Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions • Create and debug simple programs • Use logical reasoning to predict the behaviour of simple programs 	<ul style="list-style-type: none"> • 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
Information Technology	<ul style="list-style-type: none"> • Use technology purposefully to create, organise, store, manipulate and retrieve digital content 	<ul style="list-style-type: none"> • 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 programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information
Digital Literacy	<ul style="list-style-type: none"> • Recognise common uses of information technology beyond school • Use technology safely and respectfully, keeping personal information private • Identify where to go for help and support when they have concerns about content on the internet or other online technologies 	<ul style="list-style-type: none"> • Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact

Progression Overview: National Curriculum Statement and Skills

KS1	Computer Science			Information Technology	Digital Literacy	
National Curriculum	Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions	Create and debug simple programs	Use logical reasoning to predict the behaviour of simple programs	Use technology purposefully to create, organise, store, manipulate and retrieve digital content	Recognise common uses of information technology beyond school	Use technology safely and respectfully, keeping personal information private Identify where to go for help and support when they have concerns about content on the internet or other online technologies
Year 1	Children understand that an algorithm is a set of instructions used to solve a problem or achieve an objective. They know that a computer program turns an algorithm into code that the computer can understand	Children can work out what is wrong with a simple algorithm when the steps are out of order, e.g. The Wrong Sandwich in Purple Mash and can write their own simple algorithm, e.g. Colouring in a Bird activity. Children know that an unexpected outcome is due to the code they have created and can make logical attempts to fix the code, e.g. Bubbles activity in 2Code	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, for example, interpret where the turtle in 2Go challenges will end up at the end of the program	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, use Purple Mash 2Quiz example (sorting shapes), 2Code design mode (manipulating backgrounds) or using pictogram software such as 2Count	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 such as their My Work folder on Purple Mash.
Year 2	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, e.g. Debug Challenges: Chimp. Children's program designs display 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 demonstrate an ability to organise data using, for example, a database such as 2Investigate and can retrieve specific data for conducting simple searches. Children are able to edit more complex digital data such as music compositions within 2Sequence. 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 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, e.g. 2Publish example template. Children make links between technology they see around them, coding and multimedia work they do in school e.g. animations, interactive code and programs.	Children know the implications of inappropriate online searches. Children begin to understand how things are shared electronically such as posting work to the Purple Mash display board. They develop an understanding of using email safely by using 2Respond activities on Purple Mash and know ways of reporting inappropriate behaviours and content to a trusted adult.

LKS2	Computer Science				Information Technology		Digital Literacy
National Curriculum	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 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 concerns about content and contact
Year 3	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'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, repetition and use of timers. They make good attempts to 'step through' more complex code in order to identify errors in algorithms and can correct this. e.g. 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 2Email. 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 (2Question), using software such as 2Graph. Children can consider what software is most appropriate for a given task. They can create purposeful content to attach to emails, e.g. 2Respond.	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 such as 2Email in Purple Mash. They know more than one way to report unacceptable content and contact
Year 4	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'. e.g. 2Code.	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. In programs such as Logo, they can 'read' programs with several steps and predict the outcome accurately	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 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 such as 2Connect and 2Publish+. Children share digital content within their community, i.e. using Virtual Display Boards	Children can explore key concepts relating to online safety using concept mapping such as 2Connect. They can help others to understand the importance of online safety. Children know a range of ways of reporting inappropriate content and contact.

UKS2	Computer Science				Information Technology		Digital Literacy
National Curriculum	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 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 concerns about content and contact
Year 5	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, e.g. the use of tabs to organise code and the naming of variables.	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 can select the most appropriate form of online communications contingent on audience and digital content, e.g. 2Blog, 2Email, Display Boards.	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 using 2Code. 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, i.e. 2Blog, Display Boards and 2Email.	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.
Year 6	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	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	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. 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	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. 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	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.	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	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.

	<p>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 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.</p>	<p>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 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.</p>		<p>school.</p>	<p>2Blog. They are able to use criteria to evaluate the quality of digital solutions and are able to identify improvements, making some refinements.</p>	<p>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.</p>	
--	---	---	--	----------------	--	--	--



KS1 and KS2: Computing

Key Skills: Code, Connect, Communicate and Collect

Year 1	<ul style="list-style-type: none">• Know how to switch a range of digital devices (computer/laptops) on and off• Load programs (office, apps) with support/open and close apps• Use a mouse pad to navigate an age appropriate website/programme• Use a mouse pad to select/drag/position an object or window to talk about what they are doing with Computers/Digital media using appropriate vocabulary
Year 2	<ul style="list-style-type: none">• Develop an awareness of keyboard layout and use of a mouse e.g. use the mouse or arrow keys to insert words and sentences• Know backspace/undo• Know shift for capital letters and enter/upload• Change font/size colour and style of text• Use two hands when typing• Log on and off digital devices• Use navigation skills to access appropriate parts of a website/simple program/app
Year 3	<ul style="list-style-type: none">• Know how to upload from digital devices and the Internet to a shared space (Class folders/children's folder)• Know that their work can be accessed by logging onto their folder/network area• Open, edit and save their work in their own space• Use touch typing with both hands in the correct place on a keyboard• Know how to insert, cut, copy and paste• Use ctrl+v and ctrl+c to paste• Use 'save as' to create another version of their work• Develop further basic drafting skills: insert words or sentences; centre titles; change font, font size, colour; practise touch typing
Year 4	<ul style="list-style-type: none">• Use online dictionary/thesaurus• Use ctrl+alt+prntscrn to take a picture of the whole screen and paste it into paint to adapt it• Use windows snipping tool to capture and annotate work• Continue to practice touch typing• Use more than two fingers to type• Develop further basic drafting and editing skills• Edit and top copy literacy work using Word/PowerPoint/Publisher• Use spell checker• Delete, insert and replace text using mouse or arrow keys

Year 5	<ul style="list-style-type: none">• Use an online dictionary/thesaurus to search out level specific grammar and vocabulary independently• Use a variety of techniques to save and annotate on screen projects (screenshots/snipping)• Find, save, crop and edit images to suit needs of projects• Continue to practise touch typing and use several fingers when typing• Use a spellchecker and grammar checker to ensure consistency throughout work
Year 6	<ul style="list-style-type: none">• Continue to build on Year 5 Key Skills• Select suitable software to edit and redraft written work• Use a variety of keyboard shortcuts to improve efficiency on computing systems



KS1 and KS2: Computing

Computer Science

	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Knowledge and skills	<ul style="list-style-type: none"> • Understand that an algorithm is a set of instructions • Know that a computer program turns an algorithm into code that the computer can understand • Be able to work out what is wrong when the steps are out of order in instructions • Know that an unexpected outcome is due to the code they have created • Make logical attempts to fix the code • Read code one line at a time and make a good guess about what is going to happen in a program 	<ul style="list-style-type: none"> • Explain that an algorithm is a set of instructions to complete a task • Carefully plan an algorithm so it will work when made into code • Design a simple program to achieve a specific purpose • Find and correct errors in a program ('debug') • Say what will happen in a program • Identify something in a program that has an action or effect 	<ul style="list-style-type: none"> • Make a real life situation into an algorithm for a program • Design an algorithm carefully: think about what it is needed to do and how this can be turned into code • Identify an error in a program and fix it • Experiment with timers in a program • Identify the difference in using a timer command rather than a repeat command in a code • Know that a variable stores information while a program is running • Identify 'if' statements, repetition and variables • Read programs with several steps and predict what it will do • Identify the different ways that the internet can be used for communication • Use email to respond to others appropriately and attach files 	<ul style="list-style-type: none"> • Turn a real life situation to solve into an algorithm, using a design that shows how can accomplish this in code • Use repetition in a code e.g. a loop that continues until a condition is met • Use timers within a program designs more accurately to create repetition effects • Use selection (decision) in programming e.g. use an 'if' statement • Use variables within a program and know how to change the value of variables • Use user inputs and output features such as 'print to screen' • Identify errors in my code by using different methods and fixing them • Read programs that contain several steps and predict the outcomes with increasing accuracy • Recognise the main component parts of hardware which allow computers to join and form a network • Understand that network and communication components can be found in many different devices which allow them to join the internet 	<ul style="list-style-type: none"> • Make more complex real-life problems into algorithms for a program • Test and debug programs as go along • Convert (translate) algorithms that contain sequence, selection and repetition into code that works • Use sequence, selection, repetition and some other coding structures in code • Organise code carefully-naming variables and using tabs- to enable efficient debugging • Use logical methods to identify the cause of any bug with support to identify the specific line of code • Know the importance of computer networks and how they help solve problems and enhance communication • Recognise the main dangers that can be perpetuated via computer networks • Explain what personal information is and know strategies for keeping this safe • Use the most appropriate form of online communication according to the digital content 	<ul style="list-style-type: none"> • Turn a complex programming task into an algorithm • Identify the important aspects of a programming task (abstraction) • Decompose important aspects of a programming task in a logical way, identifying appropriate coding structures that would work • Test and debug a program as work on it and use logical methods to identify a cause of a bug • Identify a specific line of code that is causing a problem in a program and attempt to fix it • Translate algorithms that include sequence, selection and repetition into code and nest these structures within each other • Use inputs and outputs within coded programs such as sound, movement and buttons and represent the state of an object • Understand a program in parts and make logical attempts to put the separate parts together in an algorithm to explain the program as a whole • Explain the difference between the internet and the World Wide Web • Explain what a WAN and a LAN is and describe the process of how access to the internet in school is possible
Sticky Knowledge	<ul style="list-style-type: none"> • Know what an algorithm is • Create and follow their own simple set of instructions 	<ul style="list-style-type: none"> • Understand the need for repeat testing and debugging • Plan and use algorithms in a program 	<ul style="list-style-type: none"> • Design and write a program that simulates a physical system • Know why it is important to save their work after each functioning iteration of the program made 	<ul style="list-style-type: none"> • Use an algorithm when making a simulation of an event on the computer • Understand what a variable is 	<ul style="list-style-type: none"> • Create a playable, competitive game – designing setting, character and sounds and including clear instructions 	<ul style="list-style-type: none"> • Use coding concepts of functions, two way selection (if/else statements) and repetition in conjunction with one another to code a game

Vocabulary	Action, algorithm, arrow, backwards, challenge, code, command, computer, debug, direction, event, execute, forward, input, instruction, left turn, object, output, program, properties, run, rewind, right turn, scale, scene, sound, undo, when clicked.	Action, algorithm, background, button, collision detection, debug / debugging, design mode, event, key pressed, nesting, object, predict, properties, run, scale, scene sequence, sound, test, text, timer, when clicked / swiped.	Action, alert, algorithm, background, blocks of command, button, collision detection, command, debug / debugging, develop, event execute, flowchart, nesting, object, output, plan, predict, procedure, properties, repeat, run, sequence, scene, sound, test, timer, values.	Action, alert, algorithm, background, blocks of command, button, collision detection, code block, command, co-ordinates, debug / debugging, develop, event execute, flowchart, if, if/else nesting, number variable, object types, prompt, properties, repeat until, run, selection, sequence, scene, sound, test, timer, values, variable values Logo, BK, FD, RT, LT, Repeat, SetPC, SETPS, PU, PD	Action, abstraction, alert, algorithm, background, blocks of command, button, called collision detection, code block, command, co-ordinates, debug / debugging, decomposition, developer, event execute, flowchart, function, if, if/else nesting, object, physical system launch command, nesting, number variable, predict, procedure, prompt, properties, repeat until, run, scene, score selection, sequence, simplify, simulation sound, tab, test, timer, values, variable values Animation, Computer game, customise, evaluation, image, instructions, interactive, screenshot, texture, perspective, playability	Action, alert, algorithm, background, blocks of command, button, called, collision detection, code block, command, concept map, co-ordinates, debug / debugging, decomposition, developer, event, execute, flowchart, function, get input if, if/else launch command, nesting, number variable, object, predict, procedure, prompt, properties, repeat until, run, scene score selection, sequence, scene, simplify, simulation sound, string tab, timer, user input, values.
-------------------	---	--	---	---	---	---



KS1 and KS2: Computing

Information Technology:

	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Knowledge and skills	<ul style="list-style-type: none"> Sort sound, pictures and text Add sound, pictures and text to a program Change content on a file such as text, sound and images Name a piece of work Save a piece of work Find a piece of work 	<ul style="list-style-type: none"> Organise data – e.g. using a database Find data using specific searches Use several programs to organise information e.g. using binary trees or spreadsheets Edit digital data e.g. data in music software composition Name, save and find a piece of work Include photos, text and sound in creations 	<ul style="list-style-type: none"> Carry out searches to find digital content on a range of online systems e.g. within Purple Mash or on an internet search engine Collect data and input it into software Analyse data using features within software to help e.g. formula in spreadsheets Present data and information using different software e.g. branching database or graphing tool Consider the most appropriate software to use when given a task 	<ul style="list-style-type: none"> Understand the purpose of a search engine and the main features within it Look at information on a webpage and make predictions about the accuracy of information contained within it Create and improve solutions to a problem based on feedback Review solutions that others have created, using a checklist of criteria Work collaboratively to create content and solutions Share digital content using a variety of applications: blog, email (across units) 	<ul style="list-style-type: none"> Search precisely when using a search engine – know that adding additional words or removing words can help find better results Explain in detail how accurate, safe and reliable the content is on a webpage Make appropriate improvements to digital work created Comment on how successful a digital solution is that have created Work collaboratively with others creating solutions to problems using appropriate software Use collaborative modes (e.g. within 2Connect) to work with others and share 	<ul style="list-style-type: none"> Use filters when searching for digital content Explain in detail how accurate and reliable a web page and its content is Compare a range of digital content sources and rate them in terms of content, quality and accuracy Consider the intended audience carefully when digital content is made and designed Design and create own online blogs Use criteria to evaluate the quality of own and others digital solutions suggesting refinements
Sticky Knowledge	<ul style="list-style-type: none"> Add to a previously saved story online Use a spreadsheet to count items and give images a value 	<ul style="list-style-type: none"> Create own tune digitally and upload sounds and change them to enhance their music Solve mathematical puzzles using a spreadsheet 	<ul style="list-style-type: none"> Create own branching database – choosing images and questions themselves Touch type using both hands in the correct place on a keyboard 	<ul style="list-style-type: none"> Make practical use of a spreadsheet to help to plan actions (e.g. budgeting) Compose a piece of music incorporating a variety of notes, different pitch and creating their own rhythms 	<ul style="list-style-type: none"> Use formulae, including the advanced mode, in producing a spreadsheet Be able to insert, alter and edit images within a document 	<ul style="list-style-type: none"> Create a spreadsheet to model a real life situation and come up with solutions that can be applied to real life Change the visual properties of an online blog to appeal to the targeted audience
Vocabulary	<p>Animation, Arrow keys, backspace key, cells, clipart, collate, columns, count tool, Criteria, cursor, data, delete key, e-book, file, font, image toolbox, lock tool, move cell tool, pictogram, rows, sort, sound effect, speak tool, spreadsheet.</p>	<p>Animated, audience Avatar, Arrow keys, backspace key, Binary tree, BPM (beats per minute) cells, clipart, collate, columns, composition, concept map, copy and paste, count tool, Criteria, cursor, data, delete key, e-book, equals tool, file, font, image toolbox, instrument, internet lock tool, move cell tool, node, pictogram, presentation, quiz, rows, search, search engine share, sort, sound effect, soundtrack, speak tool, spreadsheet, template, tempo, volume</p>	<p>Animated, audio Avatar, Arrow keys, advance mode backspace key, bar chart, block graph, Binary tree), branching database, bottom row keys, cells, clipart, collate, columns, concept map, copy and paste, count tool, Criteria, cursor, data, database delete key, design template, e-book, entrance animation, equals tool, file, field font, graph, home row keys, image toolbox, internet lock tool, line graph, media move cell tool, node, pictogram, pie chart, presentation, presentation program, posture, question, quiz, rows, search, search engine share, simulation, slide, slideshow, sort, sound effect, stock image, spin tool, spreadsheet, space bar, template, text box, text formatting, top row keys, transition, word art.</p>	<p>Advance mode, animation, average function, background, BK, bold, cells, charts, copy and paste, columns, CPU, dynamics, Easter Egg, equals too, FD, flipbook, frame, font, formula, formula wizard, graphics card, internet, internet browser, italics, keyboard and mouse, logo, LT, melody, monitor, Motherboard, move cell tool, network card, onion skinning, PD, pitch, play, PU, pulse, RAM, random tool, repeat, rhythm, rippler, rows, RT, search, search engine, set PC, set PS, speakers, spin tool, spoof website, sound, spreadsheet, stop motion, tempo, texture, timer, underline, video clip, website</p>	<p>2d, 3d, Advance mode, animation, arrange, audience, Average function, avatar, binary tree (branching database), cells, charts, columns, collaborative, computer aided design (CAD) computer game, concept, concept map, connection, copy and paste, copyright, cursor, customise, data, database, document, equals tool, evaluation, find, font, formula, formula wizard, group, image, instructions, interactive, merge cells modelling, move cell tool, net, node, perspective, paragraph formatting, playability, points, polygon, random tool, record, rows, screenshot, sort, spin tool, spreadsheet, statistics, table, template, text formatting, texture, timer, viewpoint, visual, word art, word processing tool.</p>	<p>2d, 3d, Advance mode, animation, arrange, audience, Average function, avatar, binary tree (branching database), blog, blog post, blog page, cells, charts, columns, collaborative, computer aided design (CAD) computer game, concept, concept map, connection, count (how many) tool, copy and paste, copyright, cursor, customise, dice, document, equals tool, evaluation, font, formula, formula wizard, group, icon, image, instructions, interactive, internet, Local Area Network (LAN), merge cells modelling, move cell tool, net, network, network cables, node, perspective, paragraph formatting, playability, points, polygon, quiz, random tool, record, router, rows, screenshot, sort, spin tool, spreadsheet, sprite, statistics, table, template text-based adventure, text formatting, texture, timer, viewpoint, visual, Wide Area Network (WAN), wireless, word art, word processing tool, World Wide Web</p>



KS1 and KS2: Computing

Digital Literacy

	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Knowledge and skills	<ul style="list-style-type: none"> • Say what technology is • Say what examples of technology are in school • Say what examples of technology are at home • Know that a chair uses old technology and a smart phone uses new technology • Keep own login information safe • Save work on a safe place (such as 'My Work' folder) 	<ul style="list-style-type: none"> • Find information needed using a search engine • Know the consequences of not searching online safely • Share work and communicate electronically (e.g. using 2Email or the Display Boards) • Know how to report unkind behaviour and things that are upsetting online to a trusted adult • See where technology is used in school – e.g. school office, school kitchen • Understand that creations such as programs in 2Code need similar skills to the adult world 	<ul style="list-style-type: none"> • Create a secure password • Explain the importance of having a secure password and not sharing it with others • Explain the negative consequences of not keeping passwords safe and secure • Understand the importance of keeping safe online and behaving respectfully • Use communication tools such as 2Email respectfully and use good etiquette • Report unacceptable content and contact online in more than one way to a trusted adult 	<ul style="list-style-type: none"> • Have a good understanding of the online safety rules we learn at school • Demonstrate how to use different online technologies safely • Demonstrate how to use a few different online services safely • Know I have a right to privacy both on and offline • Recognise that my wellbeing can be affected by how I use technology • Report with ease any concerns with content and contact online and know immediate strategies to keep safe 	<ul style="list-style-type: none"> • Have a secure knowledge of online safety rules taught at school • Demonstrate the safe and respectful use of different online technologies and online services • Always relate appropriate online behaviour to my right to have personal privacy • Know how to not let mental wellbeing, or that of others, be affected by use of online technologies and services 	<ul style="list-style-type: none"> • Demonstrate safe and respectful use of a range of different technologies and online services • Identify more discrete inappropriate behaviours online e.g. someone who may be trying to groom them or someone else • Use critical thinking to help stay safe online • Know the value of protecting own privacy and that of others online
Sticky Knowledge	<ul style="list-style-type: none"> • Say what the importance of logging on and logging off correctly is 	<ul style="list-style-type: none"> • Be able to talk about how information put online leaves a digital footprint 	<ul style="list-style-type: none"> • Be able to open and read and email safely; and to send an email with an attachment 	<ul style="list-style-type: none"> • Identify the positive and negative influences of technology on own health and the environment 	<ul style="list-style-type: none"> • Be aware of appropriate and inappropriate text, photographs and videos and the impact of sharing these online 	<ul style="list-style-type: none"> • Have a clear idea of appropriate online behaviour and how this can protect themselves and others from possible online dangers, bullying and inappropriate behaviour
Vocabulary	Avatar, Log in, log out, my work, notification, password, save, tools, topic, username,	Display board, email, internet, search, sharing	Attachment, address book, Blog, cc, Communication, compose Concept map, email, formatting, internet, password, PEGI rating, send, username, spoof website, webpage, website	Computer virus, cookies, copyright, digital footprint, email, identity theft, malware, phishing, plagiarism, spam	Bibliography, citation, encryption, identity theft, online safety, password, plagiarism, reference, reputable, shared image smart rules.	Digital footprint, password, PEGI rating, phishing, screen time, spoof website.