

Computing

A high-quality computing education equips pupils to use computational thinking and creativity to understand and change the world. Computing has deep links with mathematics, science, and design and technology, and provides insights into both natural and artificial systems. The core of computing is computer science, in which pupils are taught the principles of information and computation, how digital systems work, and how to put this knowledge to use through programming. Building on this knowledge and understanding, pupils are equipped to use information technology to create programs, systems and a range of content. Computing also ensures that pupils become digitally literate – able to use, and express themselves and develop their ideas through, information and communication technology – at a level suitable for the future workplace and as active participants in a digital world.

The national curriculum for computing aims to ensure that all pupils:

- can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation*
- can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems*
- can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems*
- are responsible, competent, confident and creative users of information and communication technology.*

Computing skills will be taught discretely using Purple Mash and as an integrated part of a theme-based curriculum, with skills being applied in relation to each class' current topic.

....	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Computer Science	Understand that an algorithm is a set of ordered instructions that is used to solve a problem or achieve an objective. When they code, they understand that they are using an algorithm and turning it into something that the computer can interpret.	Create a computer program from an algorithm they have made, turning each step into code. They recognise that outputs from an on-screen program or physical device are directly attributed to the instructions they have coded.	Ability to plan algorithms is increasingly detailed including identifying the logical steps towards a solution with stages of decomposition of the overall task. e.g. planning the route of turtle. Their familiarity with coding structures such as sequences and timers allows them to 'read' a program.	Independently design, code, test, and debug a simple program. Their ability to decompose a task into smaller parts in order to code or interpret programs allows them greater understanding of how to achieve desired effects with code. They begin to consider efficiency when structuring code.	Consider a range of program structures when turning tasks into algorithms. They understand what simulating a physical system means and write programs that decompose these systems and model parts through code. They understand how they are using levels of abstraction and decomposition.	Understand the advantage of coding from program design documentation and do this at a deeper level of abstraction enabling them to produce accurate algorithms to code. They follow flowcharts to design and debug programs. Children use decomposition & abstraction more confidently to simplify and break down tasks.
	Create simple instructions for someone to follow, building up to creating a simple program that a computer can follow by arranging code blocks. They will recognise simple errors in a program such as the way an object is moving and attempt to debug in a purposeful way.	Create programs that include more than one object type. They can alter attributes of objects purposefully and predict and observe the effect. They vary the flow of sequences through the use of timer commands. They can program buttons to initiate events in programs. They know that unexpected errors in their programs are due to the code they have created.	Design and code simple sequential programs, using timers to create delay effects. They can use a repeating program structure as an efficient alternative to a sequence e.g. when drawing simple shapes with a turtle object.	Understand how to structure a program to use selection through the use of if and if/else statements. Their understanding of program structures enables them to code more complex algorithms that use sequencing, selection and repeat. They can explain the function of a variable and create number variables to keep track of things in a program such as scoring.	Include sequence, selection and repetition in programs. They recognise when the use of a function will improve the efficiency and readability of their code and can create and use functions in coding. They understand how both string and number variables can utilise the system memory to hold information used by the program.	Have both coding structure understanding and knowledge of a greater range of coding components that enable them to make use of variables, user inputs, cloning and hotspots to add complexity and user interactivity to their programs. They are able to translate skills and

		They attempt to debug using logical reasoning, interpreting each line of code in sequence.				understanding gained through the use of block coding programs to text based coding in Python.
	Use logical reasoning based upon their understanding of the desired algorithm and their knowledge of code blocks to predict outcomes. They focus on objects and their actions and the events that initiate these actions.	Identify the parts of a program that respond to specific events and initiate specific actions. Using a design plan or simple written algorithm, they can compare this against how a program is executing when in play mode.	Use logical reasoning and their understanding of algorithms to read and create simple flowcharts for planned or existing programs. They use these to support their efforts at debugging logically. They are beginning to consider how the nesting of code affects the program flow.	Read multi-step programs and accurately predict outcomes. They show more logical approaches to debugging, recognising how different parts of the code interact.	Have increased understanding and familiarity with a variety of coding structures allows them to 'read' code at a high level and focus logically on the parts they need to debug when something doesn't execute as designed. They use a mixture of logical reasoning and trial and testing to identify and fix bugs.	Default to a logical approach when debugging. They begin debugging at the program design and algorithm stage of tasks so their coding becomes more effective with less reliance on a try and test approach at the early stages of coding a program. They can then use a step-by-step approach to step through parts of code.
			Know the ways people can communicate with each other (digital and non-digital). They explore the use of emails and know how they can be exchanged. They are able to read, compose and send their own emails.	Develop their understanding of search engines and how they are used to find relevant information. They obtain focused results from a search query using clear keywords and refined searching techniques such as quotation marks.		Explain the function of networks and the difference between a LAN and a WAN with relevant examples. They understand the difference between the internet and the World Wide Web and can name some services as examples of each.
Information technology	Name, save and retrieve their work and follow simple instructions to access online resources. They create and combine digital content such as images, sound and text with a purpose.	Know how to access digital content they have created or have been assigned. They use a range of software to create digital content and manipulate it with purpose. For example, creating spreadsheets to organise data, merging digital artifacts from different sources into a file. They know how to save and edit work.	Children can effectively use a greater range of software tools and select tools for a purpose. They can combine digital products such as inserting music they have composed into an animation they create.	Effectively use a greater range of software tools and select tools for a purpose. They can combine digital products such as inserting music they have composed into an animation they create.	Refine their ability to select and use a growing range of software for a purpose. They create and use a range of digital products including databases, spreadsheets and word processing software. They use software to collect data, manipulate, interrogate and produce information based on a specification (query).	Use of a range of digital tools demonstrates their transferrable skills between tools. They use tools with a greater level of sophistication. For example, including their own digital artefacts within their blogs, graphs within written pieces and application of data analysis.
Digital Literacy	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 understand the importance of keeping information, such as their usernames and passwords,	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 e.g. interactive code and programs.	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	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. They recognise the value in preserving their privacy when online for their



	<p>private and actively demonstrate this in lessons.</p> <p>Children take ownership of their work and save this in their own private space.</p>	<p>Children know the implications of inappropriate online searches. Children begin to understand how things are shared. 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</p>	<p>more than one way to report unacceptable content and contact</p>			<p>own and other people's safety.</p>