National Curriculum Objectives – Computing

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.

EYFS	Technology in the Early Years can mean: • taking a photograph with a camera or tablet • searching for information on the internet • playing games on the interactive whiteboard • exploring an old typewriter or other mechanical toys • using a Beebot • watching a video clip • listening to music
KS1	Pupils should be taught to: • understand what algorithms are; • how algorithms are implemented as programs on digital devices; • understand 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 or contact on the internet or other online technologies.

KS2	**Pupils should be taught to:** • 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 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 • use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.	
Skill	Year 1	Year 2
Computer Science • Understand what algorithms are; how they are implemented as programs on digital devices; and that	Understand that an algorithm is a set of instructions used to solve a problem or achieve an objective. Know that an algorithm written for a computer is called a program. • 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. 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. • Read code one line at a time and make good attempts to envision the bigger picture of the overall effect of the program.	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 when designing a simple program so that they can be successfully converted into code. • Create a simple program that achieves a specific purpose. 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. • 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

programs execute by following precise and unambiguous instructions. • Create and debug simple programs. • Use logical reasoning to predict the behaviour of simple programs.	Interpret (for example where the turtle in 2Go challenges will end up at the end) Computer Science of the program.	
Information technology • Use technology purposefully to create, organise, store, manipulate and retrieve digital content.	Sort, collate, edit and store simple digital content e.g. 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.	Demonstrate an ability to organise data using, for example, a database such as 2Invesitigate and can retrieve specific data for conducting simple searches. Edit more complex digital data such as music compositions within 2Sequence. Confident when creating, naming, saving and retrieving content. Use a range of media in their digital content including photos, text and sound.
Digital Literacy • Recognise common uses of information technology beyond school.	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. Digital Literacy 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	Effectively retrieve relevant, purposeful digital content using a search engine. Apply their learning of effective searching beyond the classroom. Share this knowledge, e.g. 2Publish example template. Make links between technology they see around them, coding and multimedia work they do in school e.g. animations, interactive code and programs. • Know the implications of

• 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	ownership of work and save in their own private space their My Work folder on Purple Mash.	inappropriate online searches. Begin to understand how things are shared electronically such as posting work to the Purple Mash display board. 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.
Skill	Year 3	Year 4
Computer Science • 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.	Turn a simple real life situation into an algorithm for a program by deconstructing it into manageable parts. Design shows that they are thinking of the desired task and how this translates into code. 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. Begin to understand the difference in the effect of using a timer command rather than a repeat command when creating repetition effects. Understand how variables can be used to store information while a program is executing. • Designs for their programs show 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. Good attempts to 'step through' more complex code in order to identify errors in algorithms and can correct this. e.g. traffic light algorithm in 2Code. In programs such as Logo, 'read' programs with several steps and predict the outcome accurately. • List a range of ways that the internet can be used to provide different	Turn a real-life situation into an algorithm - the design shows thought for the required task and how to accomplish this in code using coding structures for selection and repetition. Make more intuitive attempts to Computer Science debug their 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 they design in their programs. As well as understanding how variables can be used to store information while a program is executing, use and manipulate the value of variables. Make use of user inputs and outputs such as 'print to screen'. e.g. 2Code. • Designs for their programs show thought for 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. e.g. traffic light algorithm in 2Code. In programs such as Logo, they can 'read' programs with several steps and predict

• 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.	methods of communication. Use some of these methods of communication, e.g. being able to open, respond to and attach files to emails using 2Email. Describe appropriate email conventions when communicating in this way.	the outcome 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 is Computer Science improving.
Information technology • 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	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. • 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. Consider what software is most appropriate for a given task. Create purposeful content to attach to emails, e.g. 2Respond.	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 such as 2Connect and 2Publish+. Share digital content within their community, i.e. using Virtual Display Boards.

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.		
Digital Literacy • Use technology safely, respectfully and responsibly; recognise acceptable/ unacceptable behaviour; identify a range of ways to report concern about content and contact.	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. Understand the importance of staying safe and the importance of their conduct when using familiar communication tools such as 2Email in Purple Mash. Know more than one way to report unacceptable content and contact.	Explore key concepts relating to online safety using concept mapping such as 2Connect. Help others to understand the importance of online safety. Know a range of ways of reporting inappropriate content and contact.
Skill	Year 5	Year 6
Computer Science • Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by	Attempt to turn more complex real-life situations into algorithms for a program by deconstructing it into manageable parts. Test and debug 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. • 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. Combining sequence, selection and repetition with other coding	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

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.	structures to achieve their algorithm design. • 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 but are also aware of the main dangers. Recognise what personal information is and can explain how this can be kept safe. Select the most appropriate form of online communications contingent on audience and digital content, e.g. 2Blog, 2Email, Display Boards.	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. • 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. • Understand and can explain in some depth 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.
Information technology • Use search technologies	• 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. • Make appropriate improvements to digital solutions based on feedback received and	• 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 are able to rate them in terms of content quality and accuracy. Use critical thinking

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 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.	can confidently comment on the success of the solution. e.g. creating their own program to meet a design brief using 2Code. 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, i.e. 2Blog, Display Boards and 2Email.	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 • Use technology safely, respectfully and responsibly; recognise acceptable/ unacceptable behaviour; identify a range of ways to report concern about content and contact.	Have 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. Implicitly relate appropriate online behaviour to their right to personal privacy and mental wellbeing of themselves and others.	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 activities. Recognise the value in preserving their privacy when online for their own and other people’s safety.