

Computer science has been broken down into three strands: Computational Thinking, Programming and Computer Networks. Computational Thinking is all about solving problems effectively with or without a computer. Computational thinking is about looking at a problem in a way in which a computer can help us to solve it. This is a two-step process: 1. First, we think about the sequence of steps (an algorithm) needed to solve a problem 2. Then, we use our technical skills to get the computer working on the problem as we implement our algorithm as code. As demonstrated with Dr Chips' support videos, a lot of these objectives can be applied across the curriculum. Programming is one application of computational thinking. Learners will write algorithms and implement these as code. They also need to be able to find mistakes and fix them (debugging.) Once learners have created a program they need to learn to evaluate and look at different ways to achieve the same goal and which method is most appropriate. As learners get older the programs they write will become more complex using a range of constructs such as sequence, selection, repetition and variables in their programs. KS 2 pupils also require knowledge of networks, such as the Internet, work and how searches are performed.

Computational Thinking

Year Group	NC Objectives	Skills/Knowledge	Apps and Links
EYFS	Children recognise that a range of technology is used in places such as homes and schools. They select and use technology for particular purposes.	<ul style="list-style-type: none"> • I can follow simple oral instructions • I can spot simple patterns, such as similarities and differences • I can sequence simple familiar tasks 	<u>MrPICT.com Resources</u>
1	<ul style="list-style-type: none"> • Co2/1.1 understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions 	<ul style="list-style-type: none"> • I understand what algorithms are • I can write simple algorithms • I understand the sequence of algorithms is important • I can debug simple algorithms 	<u>MrPICT.com Resources</u>
2	<ul style="list-style-type: none"> • Co2/1.2 create and debug simple programs • Co2/1.3 use logical reasoning to predict the behaviour of simple programs 	<ul style="list-style-type: none"> • I can write algorithms for everyday tasks • I can use logical reasoning to predict the outcome of algorithms • I understand decomposition is breaking objects/processes down • I can debug algorithms 	<u>MrPICT.com Resources</u>

Computational Thinking

Year Group	NC Objectives	Skills/Knowledge	Apps and
3	<ul style="list-style-type: none"> Co2/1.1 design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts 	<ul style="list-style-type: none"> I can create algorithms for my programming projects I can decompose projects (such as an animation) into steps to create an algorithm I understand abstraction is focusing on important information I can identify patterns in an algorithm 	MrPICT.com Resources
4	<ul style="list-style-type: none"> Co2/1.2 use sequence, selection, and repetition in programs; work with variables and various forms of input and output Co2/1.3 use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs 	<ul style="list-style-type: none"> I can use abstraction to focus on what's important in my design I can write more precise algorithms for use when programming I can use simple selection and repetition in algorithms I can use logical reasoning to detect and correct errors in programs 	MrPICT.com Resources
5	<ul style="list-style-type: none"> Co2/1.4 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 	<ul style="list-style-type: none"> I can solve problems by decomposing them into smaller parts I can use selection in algorithms I can use logical reasoning to explain how a variety of algorithms work I can evaluate the effectiveness of algorithms 	MrPICT.com Resources
6		<ul style="list-style-type: none"> I can write precise algorithms for use when programming I can decompose a design or code to focus on specific parts I can use abstraction to hide complexity in my design or code I can recognise and make use of patterns in my design and code I can critically evaluate my work and suggest improvements 	MrPICT.com Resources

Coding and Programming

<u>Year Group</u>	<u>NC Objectives</u>	<u>Skills/Knowledge</u>	<u>Apps and Links</u>
EYFS	Children recognise that a range of technology is used in places such as homes and schools. They select and use technology for particular purposes.	<ul style="list-style-type: none"> • I can use a mouse, touch screen or appropriate access device to target and select options on screen • I can input a simple sequence of commands to control a digital device with support 	Beebot, Daisy The Dinosaur
1	<ul style="list-style-type: none"> • Co2/1.1 understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions 	<ul style="list-style-type: none"> • I can create a simple program on a digital device e.g. Bee Bot • I can use sequence in programs • I can locate and fix bugs in my program 	Beebot, <u>Scratch Jnr.</u> , Kodable,
2	<ul style="list-style-type: none"> • Co2/1.2 create and debug simple programs • Co2/1.3 use logical reasoning to predict the behaviour of simple programs 	<ul style="list-style-type: none"> • I understand programs follow precise instructions • I can create programs using different digital devices E.g. Bee Bot or ScratchJr on a tablet • I can debug programs of increasing complexity • I can use logical reasoning to predict the outcome of simple programs 	Beebot, <u>Scratch Jnr.</u> , Kodable, Tynker,

Coding and Programming

Year Group	NC Objectives	Skills/Knowledge	Apps and Links
3	<ul style="list-style-type: none"> Co2/1.1 design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts 	<ul style="list-style-type: none"> I can design a program I can create a program using a design I can create a sequence of code I can work with different inputs I can evaluate my program 	Beebot, Scratch Jnr , Kodable, Tynker, Scratch 3 , Hopscotch, Swift Playgrounds ,
4	<ul style="list-style-type: none"> Co2/1.2 use sequence, selection, and repetition in programs; work with variables and various forms of input and output 	<ul style="list-style-type: none"> I can use repetition in programs I can use simple selection in programs I can work with different outputs I can use logical reasoning to systematically detect and correct errors in programs 	Beebot, Scratch Jnr , Kodable, Tynker, Scratch 3 , Hopscotch, Swift Playgrounds ,
5	<ul style="list-style-type: none"> Co2/1.3 use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs Co2/1.4 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 	<ul style="list-style-type: none"> I can create programs by decomposing them into smaller parts I can use a variety of selection commands in programs I can use conditions in repetition commands I can work with variables I can create programs that control or simulate physical systems I can evaluate my work and identify errors 	Beebot, Scratch Jnr , Kodable, Tynker, Scratch 3 , Hopscotch, Swift Playgrounds ,
6	<ul style="list-style-type: none"> Co2/1.4 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 	<ul style="list-style-type: none"> I can use a range of sequence, selection and repetition commands to implement my design I can identify the need for, and work with, variables I can create procedures to hide complexity in programs I can identify and write generic code for use across multiple projects I can critically evaluate my work and suggest improvements I can identify and use basic HTML tags (See Computer Networks objectives) 	Beebot, Scratch Jnr , Kodable, Tynker, Scratch 3 , Hopscotch, Swift Playgrounds ,

Computer Networks (KS2 only)

<u>Year Group</u>	<u>NC Objectives</u>	<u>Skills/Knowledge</u>	<u>Apps and Links</u>
3	<ul style="list-style-type: none"> • 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 	<ul style="list-style-type: none"> • I understand that computers in a school are connected together in a network • I understand why computers are networked • I understand the difference between the Internet and the World Wide Web (WWW) 	MrPICT.com Resources
4		<ul style="list-style-type: none"> • I understand that servers on the Internet are located across the planet • I understand how email is sent across the Internet • I understand how the Internet enables us to collaborate 	MrPICT.com Resources
5		<ul style="list-style-type: none"> • I understand how we view web pages on the Internet • I use search technologies effectively • I understand that web spiders index the web for search engines • I appreciate how pages are ranked in a search engine 	MrPICT.com Resources
6		<ul style="list-style-type: none"> • I understand what HTML is and recognize HTML tags • I know a range of HTML tags and can remix a web page • I can create a webpage using HTML 	MrPICT.com Resources

Sound

<u>Year Group</u>	<u>NC Objectives</u>	<u>Skills/Knowledge</u>	<u>Apps and Links</u>
3	Co2/1.6 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.	<ul style="list-style-type: none"> • I can create and edit purposeful compositions using music software to create mood or a certain style • I can experiment with live loops to create a song. 	<u>Seesaw</u> , <u>Voice Memos</u> , <u>Garageband</u> , <u>Anchor</u> , <u>Keezy</u> ,
4		<ul style="list-style-type: none"> • Edit sound effects for a purpose. • Create a simple four chord song following the correct rhythm. • I can record a radio broadcast or audiobook. 	<u>Seesaw</u> , <u>Voice Memos</u> , <u>Garageband</u> , <u>Anchor</u> , <u>Keezy</u> ,
5		<ul style="list-style-type: none"> • Add voice over and edit sound clips (volume, pitch, fade, effect) to create a podcast. • Create a remix of a popular song. 	<u>Seesaw</u> , <u>Voice Memos</u> , <u>Garageband</u> , <u>Anchor</u> , <u>Keezy</u> ,
6		<ul style="list-style-type: none"> • Add voice over and edit sound clips (volume, pitch, fade, effect) to use in a film or radio broadcast (podcast) • Compose a soundtrack that can be added to a film project. 	<u>Seesaw</u> , <u>Voice Memos</u> , <u>Garageband</u> , <u>Anchor</u> , <u>Keezy</u> ,