

## Computing Overview – EYFS, Key Stage 1 and Key Stage 2

		Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
EYFS	<b>Reception</b>	Learning how to use equipment - listening centre, bee bots, computers and mice, interactive whiteboard FOCUS: Using iPads to take photographs of emotions (Jigsaw link)  (3 Hours)	Walkie Talkies for messages (3 Hours)	Mister Maker app to make Piet Mondrian inspired art  (3 Hours)	Beebots to plan a route/journey (3 Hours)	Arcimboldo art (mouse control) (3 Hours)	Old and new toys (3 Hours)
		Key Stage 1	<b>Year 1</b>	Using a computer (simple)  (6 Hours)	Beebots  (6 Hours)	Creating a program  (6 Hours)	
<b>Year 2</b>	We are astronauts  (6 Hours)		We are game testers (6 Hours)	We are zoologists  (6 Hours)			
Lower Key Stage 2	<b>Year 3</b>	We are presenters  (6 Hours)	We are programmers.  (6 Hours)	We are communicators  (6 Hours)			
	<b>Year 4</b>	We are software designers  (6 Hours)	We are toy designers  (6 Hours)	We are musicians  (6 Hours)			
Upper Key Stage 2	<b>Year 5</b>	We are game developers  (6 Hours)	We are cryptographers  (6 Hours)	We are artists  (6 Hours)			
	<b>Year 6</b>	We are app planners (6 Hours)	We are interface designers (6 Hours)	We are APP developers (6 Hours)			

## Early Years Foundation Stage - Related to Computing

### Physical Development

### The World

#### Moving and Handling Early Learning Goal

Children show good control and co-ordination in large and small movements. They move confidently in a range of ways, safely negotiating space. They handle equipment and tools effectively, including pencils for writing

#### Technology Early Learning Goal

Children recognise that a range of technology is used in places such as homes and schools. They select and use technology for particular purposes.

	All About Me!	Let's Pretend	Animals	Journeys	Traditional Tales	Changes
Year R	<ul style="list-style-type: none"> <li>learning how to use equipment - listening centre, bee bots, computers and mice, interactive whiteboard</li> <li>FOCUS: Using iPads to take photographs of emotions (Jigsaw link)</li> </ul>	<ul style="list-style-type: none"> <li>Using walkie talkies for messages.</li> </ul>	<ul style="list-style-type: none"> <li>Digital art – using apps to create art.</li> </ul>	<ul style="list-style-type: none"> <li>Beebots to plan a route/journey.</li> </ul>	<ul style="list-style-type: none"> <li>Arcimboldo art – mouse control.</li> </ul>	<ul style="list-style-type: none"> <li>Exploring old and new toys.</li> </ul>

## National Curriculum – Computing

### Purpose of study

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.

### Aims

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.

## National Curriculum - Key stage 1

Pupils should be taught to:

- 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 or contact on the internet or other online technologies.

	To Using a computer (simple)pic	Beebots	Creating a program
Year 1	To use technology safely and respectfully, keeping personal information private	<ul style="list-style-type: none"> <li>Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions.</li> </ul>	Create and debug simple programs
Year 2	<p><b>We are astronauts</b></p> <p>To use technology safely and respectfully, keeping personal information private.</p>	<p><b>We are game testers</b></p> <ul style="list-style-type: none"> <li>Create and debug simple programs.</li> <li>Use logical reasoning to predict the behaviour of simple programs.</li> <li>Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions.</li> </ul>	<p><b>We are zoologists</b></p> <ul style="list-style-type: none"> <li>Use technology purposefully to create, organise, store, manipulate and retrieve digital content recognise common uses of information technology beyond school.</li> </ul>



## National Curriculum - Key stage 2

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.

	We are Presenters	We are Programmers	We are Communicators
Year 3	<ul style="list-style-type: none"> <li>• I can work a video camera.</li> <li>• I can record footage to use in my video.</li> <li>• I can upload and edit my footage on a computer.</li> <li>• I can record an audio commentary for my video.</li> <li>• I can study sports programmes to learn how they are filmed.</li> <li>• I can record high quality footage.</li> <li>• I can record an audio commentary with useful information in it.</li> <li>• I can export my final video in a standard format.</li> <li>• I can look at my footage and decide what does and doesn't work.</li> <li>• I can record original and interesting footage.</li> <li>• I can use and explain data in my audio commentary.</li> </ul> <p>I can use more difficult editing tools, e.g. creating transitions.</p>	<ul style="list-style-type: none"> <li>• I can create a storyboard for an animation.</li> <li>• I can include action and dialogue in my storyboard.</li> <li>• I can write a computer program for an animation.</li> <li>• I can put Scratch blocks in the right order.</li> <li>• I can correct mistakes in my program.</li> <li>• I can create sound and graphics for my animation.</li> <li>• I can explain how my storyboard and program are linked.</li> <li>• I can use a repeat block in my program.</li> <li>• I can find and correct 'bugs' in my program.</li> <li>• I can upload my animation to the Scratch website.</li> </ul> <p>I can get ideas from the Scratch website.</p>	<ul style="list-style-type: none"> <li>• I can see how email and video conferencing work on the internet.</li> <li>• I can use email and video conferencing to communicate.</li> <li>• I can write an email and speak on video to communicate with others.</li> <li>• I can follow my school's rules and use email and video conferencing safely.</li> <li>• I can see that the internet and the web are different.</li> <li>• I can work with my partner well.</li> <li>• I can show respect for my partner's ideas.</li> <li>• I can let my teacher know if I am unsure about something in an email.</li> <li>• I can work independently with my partner to plan our work.</li> <li>• I can tell my partner what I think does and doesn't work.</li> </ul> <p>I can explain some of the dangers of emails and opening email attachments.</p>

Year 4	<b>We are Software Designers</b>	<b>We are Toy Designers</b>	<b>We are Musicians</b>
	<ul style="list-style-type: none"> <li>• I can design an interactive educational game.</li> <li>• I can develop an interactive educational game.</li> <li>• I can put Scratch blocks in the right order.</li> <li>• I can use the if/then/else block correctly.</li> <li>• I can use the random number block and use variables to work out the score.</li> <li>• I can include sound in my game.</li> <li>• I can correct mistakes in my game.</li> <li>• I can plan my own way to program my game.</li> <li>• I can use a countdown timer.</li> <li>• I can use the mouse to control my game.</li> <li>• I can explain how the algorithm of my game works.</li> </ul>	<ul style="list-style-type: none"> <li>• I can design a toy with computer-controlled input and output.</li> <li>• I can write a program to show how my toy would produce output.</li> <li>• I can use Scratch to test how input and output would work in my toy.</li> <li>• I can use Scratch to work out why my toy may not work as expected.</li> <li>• I can use Scratch to create a version of my toy with computer-controlled input and output.</li> <li>• I can use Scratch to create a version of my toy using both mouse and keyboard input.</li> <li>• I can find and correct 'bugs' in my program.</li> <li>• I can explain how I find and correct 'bugs' in my program.</li> <li>• I can work out ways around problems by breaking them into smaller steps.</li> </ul>	<ul style="list-style-type: none"> <li>• I can explain how technology can be used to create music.</li> <li>• I can use sequencing software to create a piece of music.</li> <li>• I can record my own sound samples.</li> <li>• I can mix sound samples to create a piece of music.</li> <li>• I can export the file of my piece of music in a standard, compressed format.</li> <li>• I can explain how people listen to and buy music through technology.</li> <li>• I can edit sound samples.</li> <li>• I can work on and make my piece of music better.</li> <li>• I can edit my final piece of music.</li> <li>• I can use software that uses staff notation.</li> <li>• I can compare creating a piece of music to creating a program.</li> <li>• I can respect other people's copyright.</li> </ul>

Year 5	<b>We are Games Developers</b>	<b>We are Cryptographers</b>	<b>We are Artists</b>
	<ul style="list-style-type: none"> <li>• I can create a storyboard or diagram for an algorithm for my game.</li> <li>• I can create sound and graphics in Scratch for my game.</li> <li>• I can put instructions in the right order for my game.</li> <li>• I can find mistakes in my game.</li> <li>• I can create and add music for my game.</li> <li>• I can use selection and repetition in my game.</li> <li>• I can correct mistakes in my game.</li> <li>• I can listen to my partner's ideas about my game and make it better.</li> <li>• I can add instructions to my game.</li> <li>• I can break my game into smaller parts and work on them separately.</li> </ul>	<ul style="list-style-type: none"> <li>• I can send and receive messages in Morse code and semaphore.</li> <li>• I can create and decode secret messages using the Caesar and substitution ciphers.</li> <li>• I can see how important it is to keep passwords secret.</li> <li>• I can see how secret code needs to be used sometimes when using the web.</li> <li>• I can send and receive messages in Morse code and semaphore beyond the line-of-sight.</li> <li>• I can decode a message using the Caesar cipher without knowing the letter key shift.</li> <li>• I can see how important it is to create secure, hard-to-guess passwords.</li> <li>• I can check to see if a web page is in secret code ('encrypted').</li> <li>• I can explain how Morse code and semaphore are similar and different from the internet.</li> </ul>	<ul style="list-style-type: none"> <li>• I can create a tessellating pattern.</li> <li>• I can write a program to draw a simple shapes.</li> <li>• I can create a pattern using overlapping shapes.</li> <li>• I can create a pattern using different repeated shapes.</li> <li>• I can create a computer-generated image of a landscape.</li> <li>• I can create a tessellating pattern using more complicated shapes.</li> <li>• I can use repetition in Scratch to draw a complicated geometric shape.</li> <li>• I can use the tile clone tool to create a pattern using different kinds of shapes.</li> <li>• I can create a computer-generated image of a landscape that looks good.</li> <li>• I can write blocks of script in Scratch to create a complicated geometric shape.</li> </ul>

	<ul style="list-style-type: none"> <li>• I can animate my characters by creating different graphics for them.</li> <li>• I can use variables in my game.</li> <li>• I can explain how my game works.</li> <li>• I can add comments to the script of my game</li> </ul>	<ul style="list-style-type: none"> <li>• I can explain the algorithm for the Caesar cipher.</li> <li>• I can decode a message which has used a random substitution cipher.</li> <li>• I can create a secure, hard-to-guess password.</li> <li>I can check the security certificates for a web page.</li> </ul>	<p>I can explain how computers create realistic landscapes.</p>
--	--	--	---

<p>Year 6</p>	<p><b>We are App Planners</b></p>	<p><b>We are Interface Designers</b></p>	<p><b>We are App Developers</b></p>
	<ul style="list-style-type: none"> <li>• I can find geotagged photos on a map and use GPS to find photos that have been geotagged.</li> <li>• I can see that a smartphone is a computer.</li> <li>• I can come up with interesting problems that I could solve with an app.</li> <li>• I can research apps that already exist that may solve my problem.</li> <li>• I can create and present a well-planned presentation for my app.</li> <li>• I can name and describe the inputs and outputs of smartphones.</li> <li>• I can show how apps that already exist will solve a problem.</li> <li>• I can explain how search engines order web pages in a search ('Page Rank').</li> <li>• I can explain how smartphones connect to the internet through the phone network.</li> <li>• I can explain how GPS works and how it can be used in practice.</li> <li>• I can use different types of media (e.g. video) in my app presentation.</li> </ul>	<ul style="list-style-type: none"> <li>• I can sketch my ideas for the design of my app.</li> <li>• I can create screen layouts for my app using a wire framing tool.</li> <li>• I can think about how people will use my app as I design it.</li> <li>• I can see how important it is that everyone should be able to use an app.</li> <li>• I can find media assets (e.g. buttons or backgrounds) for my app.</li> <li>• I can sketch my ideas for a user-friendly design of my app.</li> <li>• I can try to design my app so that anyone should be able to use it.</li> <li>• I can create my own media assets for my app.</li> <li>• I can explain how different parts of my app will work together.</li> <li>• I can create user-friendly screen layouts for my app using a wire framing tool.</li> <li>• I can create an attractive design to suit the way people will use my app.</li> <li>• I can follow examples of good design to make sure anyone can use my app.</li> </ul> <p>I can find and credit media assets I use from other places correctly.</p>	<ul style="list-style-type: none"> <li>• I can create and write an algorithm for my app, and convert it to a code.</li> <li>• I can keep testing and improving the algorithm to find the 'bugs' in my code.</li> <li>• I can think through and work out where mistakes are in my algorithm.</li> <li>• I can use sequence, selection, repetition and variables in my code.</li> <li>• I can think through and work out where mistakes are in my code.</li> <li>• I can listen to and act on other people's ideas to improve my code.</li> <li>• I can think through and work out how to correct mistakes in my algorithm.</li> <li>• I can use procedures in my code.</li> <li>• I can think through and work out how to correct mistakes in my code.</li> <li>• I can sort and deal with problems and new features for my app in a sensible order.</li> </ul>

