Great Corby School and Nursery

Computing Progression

The EYFS framework is structured very differently to the national curriculum as it is organised across **seven areas of learning** rather than subject areas.

- Communication and Language
- Personal, social and emotional development
- Physical development
- Literacy
- Mathematics
- Understanding the World
- Expressive art and design.

Alongside the seven areas of learning are t**he Characteristics of effective teaching and learning.**

In planning and guiding what children learn, practitioners must reflect on the different rates at which children are developing and adjust their practice appropriately.

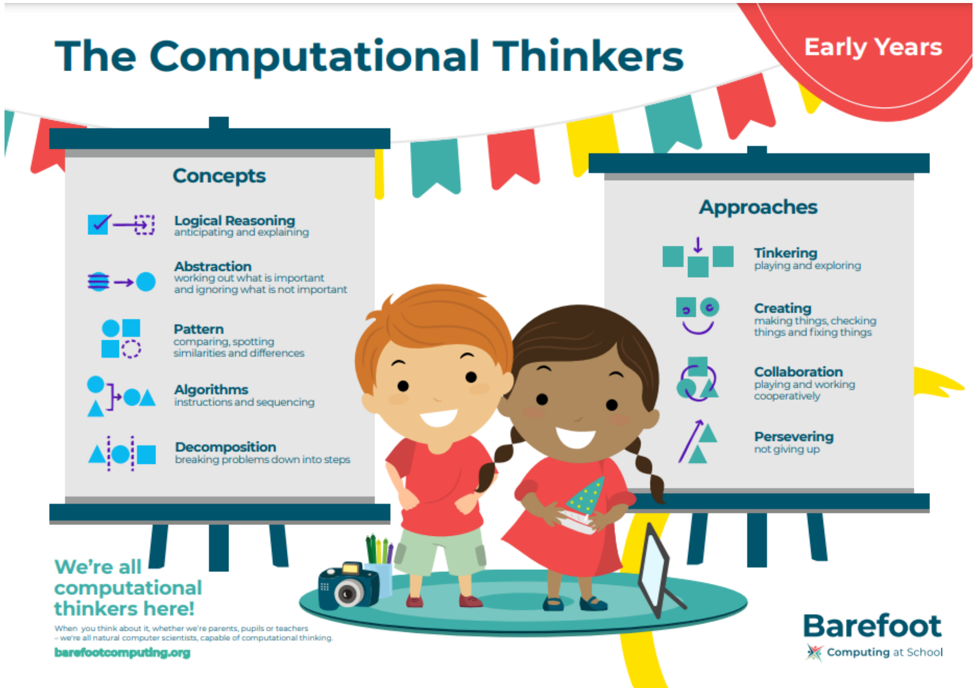
Three characteristics of effective teaching and learning are:

• *playing and exploring – children investigate and experience things, and 'have a go'*

• *ac0tive learning – children concentrate and keep on trying if they encounter difficulties, and enjoy achievements*

- *creating and thinking critically – children have and develop their own ideas, make links between ideas, and develop strategies for doing things*

Taken from Development Matters revised 2021



The Computational Thinkers

Early Years

Concepts

Logical Reasoning
anticipating and explaining

Abstraction
working out what is important and ignoring what is not important

Pattern
comparing, spotting similarities and differences

Algorithms
instructions and sequencing

Decomposition
breaking problems down into steps

Approaches

Tinkering
playing and exploring

Creating
making things, checking things and fixing things

Collaboration
playing and working cooperatively

Persevering
not giving up

We're all computational thinkers here!

When you think about it, whether we're parents, pupils or teachers – we're all natural computer scientists, capable of computational thinking.

barefootcomputing.org

Barefoot
Computing at School

The aim of this document is to help subject leaders to understand how the skills taught across EYFS feed into national curriculum subjects.

Children are given opportunities to develop their own play and independent exploration through our vision of curiosity and wonder. Communication and Language and Personal, Social and Emotional Development are intertwined in everything we do.

This document demonstrates which statements from the revised 2021 Development Matters are prerequisite skills for **computing** within the national curriculum. Although there is no specific teaching of computing in the foundation stage, we develop children's computational thinking.

The table below outlines the most relevant statements taken from the Early Learning Goals in the EYFS statutory framework and the Development Matters age ranges for Three and Four-Year-Olds and Reception to match the programme of study for **computing.**

The most relevant statements for **computing** are taken from the following areas of learning:

- The Characteristics of effective learning.
- Communication and Language
- Physical Development
- Mathematics
- Expressive Arts and Design

Computing					
Development Matters			**Vocabulary**		**Computing in Key Stage 1**
			Pattern Sequencing De-bugging Algorithms Predict First, then, after, before… Explain Perseverance	Children will be introduced to activities based around computational thinking concepts and approaches. These activities will give the children the necessary problem-solving skills needed for everyday life ***Mathematics for three and four year olds.*** •Provide patterns from different cultures, such as fabrics. Provide a range of natural and everyday objects and materials, as well as blocks and shapes, for children to play with freely and to make patterns with. •When appropriate, encourage children to continue patterns and spot mistakes. •Engage children in following and inventing movement and music patterns, such as clap, clap, stamp. •Talk about patterns of events, in cooking, gardening, sewing or getting dressed. Suggestions: • 'First', 'then', 'after', 'before' • "Every day we…" • "Every evening we…" •Talk about the sequence of events in stories. Use vocabulary like 'morning', 'afternoon', 'evening' and 'night-time', 'earlier', 'later', 'too late', 'too soon', 'in a minute'.	***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
Three and Four-Year-Olds	Personal, Social and Emotional Development	• Remember rules without an adult to remind them			
	Physical Development	• Match their developing physical skills to tasks and activities in the setting			
	Mathematics	• Talk about and identify patterns around them. •Notice and create ABAB patterns -stick, leaf, stick, leaf. •Notice and correct an error in a repeating pattern. Begin to describe a sequence of events, real or fictional, using words such as 'first', 'then…'			
	Understanding the World	• Explore how things work			
Children in Reception	Personal, Social and Emotional Development	• Show resilience and perseverance in the face of challenge. • Know and talk about different factors that support their overall health and wellbeing: sensible amounts of 'screen time'.			

	Physical Development		•Develop their small motor skills so that they can use a range of tools ideas and feelings		•Countdown to forthcoming events on the calendar in terms of number of days or sleeps. •Refer to the days of the week, and the day before or day after, ‘yesterday’ and ‘tomorrow’. ***Personal, Social and Emotional Development*** •Help them to develop problem-solving skills by talking through how they, you and others resolved a problem or difficulty. •Show that mistakes are an important part of learning and going back is trial and error not failure. •Use picture books and other resources to explain the importance of the different aspects of a healthy lifestyle. ***Mathematics in Reception*** •Challenge children to copy increasingly complex 2D pictures and patterns with these 3D resources, guided by knowledge of learning trajectories: “I bet you can’t add an arch to that,” or “Maybe tomorrow someone will build a staircase.” •Investigate how shapes can be combined to make new shapes: for example, two triangles can be put together to make a square. Encourage children to predict what shapes they will make when paper is folded. Wonder aloud how many ways there are to make a hexagon with pattern blocks. •Find 2D shapes within 3D	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.
	Mathematics		•Select, rotate and manipulate shapes to develop spatial reasoning skills. •Compose and decompose shapes so that children recognise a shape can have other shapes *within* it, just as numbers can. •Continue, copy and create repeating patterns			
	Expressive Arts and Design		• Explore, use and refine a variety of artistic effects to express their ideas feelings.			
ELG	Personal, Social and Emotional Development	Managing Staff	Be confident to try new activities and show independence, resilience, and perseverance in the face of challenge • Explain the reasons rules, know right from wrong and try to behave accordingly			
	Expressive Arts and Design	Creating with materials	Safely use and explore a variety of materials, tools and techniques, experimenting with colour, design, texture, form and function.			

	🟩	🟩			shapes, including through printing or shadow play. •Make patterns with varying rules (including AB, ABB and ABBC) and objects and invite children to continue the pattern. •Make a deliberate mistake and discuss how to fix it.	

KS1	KS2	KS3
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.	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.	Pupils should be taught to: • design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems • understand several key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem • use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions • understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming; understand how numbers can be represented in binary, and be able to carry out simple operations on binary numbers [for example, binary addition, and conversion between binary and decimal] • understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems • understand how instructions are stored and executed within a computer system; understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits • undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve

		challenging goals, including collecting and analysing data and meeting the needs of known users • create, re-use, revise and re-purpose digital artefacts for a given audience, with attention to trustworthiness, design and usability • understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct and know how to report concerns.

Digital Literacy

Information Technology

Computer Science

Computing Overview Cycle A						
	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Online Safety						
KS1 Y1 &Y2	Programming: What is an algorithm? • Explain what an algorithm is • Explain clear algorithms • Follow an algorithm • Explain what inputs and outputs are • Create an achievable program • Decompose a design into steps • Identify bugs in algorithm and how to fix them Vocabulary	Computing Systems and Networks: Improving Mouse Skills • Use computers more purposefully • Log in and navigate around a computer • Drag, drop, click and control a cursor using a mouse • Use software tools to create art on the computer Vocabulary log in log out/off	Skills Showcase: Rocket to the Moon • Use a computer to make a list • Explain the benefits of making a list on the computer • Use a basic range of tools on graphics editing software to design a rocket • Sequence instructions • Follow instructions to build their model rocket • Input data about their rockets into a table or spreadsheet	Computer Systems and Networks: What is a computer? • Name some computer peripherals and their function • Recognise that buttons cause effects • Explain that technology follows instructions • Recognise different forms of technology • Design an invention which includes	Programming: Algorithms and Debugging • Decompose a game to predict the algorithms • Give a definition for ‘decomposition’ • Write clear and precise algorithms • Create algorithms to solve problems • Use loops in their algorithms to make their code more efficient • Explain what abstraction is	Computer Systems and Networks: Word Processing • Explain which are the home row keys and how to find them for typing • Use the spacebar and backspace correctly • Type and make simple alterations to text using buttons on a word processor • Search for, import and alter appropriate images for a text document

	clear debug decomposition directions instructions code decompose device input manageable motion organise precise problem sensor order output programming robot sequence solution steps specific tasks	mouse pointer keyboard password software ctrl right click layers drag digital photograph login mouse click screen account duplicate tools menu username drag drop undo cursor	**Vocabulary** annotate components data designing digital image e-document editing folder folder instructions photo order share spreadsheet cells create debug digital content document edit evaluate input log in program robot sequence software table	inputs and outputs • Explain the role of computers in the world around them **Vocabulary** battery camera desktop digital electricity input keyboard monitor output scanner system technology buttons computer device digital recorder function invention laptop mouse paying till screen table video wires	**Vocabulary** Abstraction artificial intelligence clear data decompose key features predict algorithm bug correct debug error loop unnecessary	• Modify text in a document • Use copy and paste to copy text from one document to another • Explain what information is safe to be shared online **Vocabulary** backspace copy cut forward button home row bold copyright delete highlight home screen image italics keyboard characters keyboard navigate import keyword keyboard shortcut layout paste redo space bar text effects underline

						search text touch typing undo
LKS2 Y3 & Y4	**Computing Systems and Networks: Emailing** • Log in and out of email • Send a simple email with a subject plus ‘To’ and ‘From’ in the body of the text • Edit an email • Type in the email address correctly and send the email • Add an attachment to an email • Write an email using positive language, with an awareness of how it will make the recipient feel Vocabulary attachment cc (carbon copy) content document download	**Programming: Scratch** • Explain what some of the blocks in Scratch do • Explain what a loop is and include one in their program • Suggest possible additions to an existing program • Recognise where something on screen is controlled by code • Use a systematic approach to find bugs • Explain what an algorithm is and its purpose Vocabulary algorithm application code block debug interface	Creating Media: Video Trailers • Describe the purpose of a trailer • Create a storyboard for a book trailer • Consider camera angles when taking photos or videos • Add text to a video • Incorporate transitions between images • Evaluate their own and others’ trailers Vocabulary application clip edit fade to white film editing software import music plan slide	Creating Media: Website Design • Create a Sway with title, image and a completed first header section • Create a clear plan for their web page and beginning to create it • Create a professional looking web page with useful information and a clear style • Create a clear plan by referring back to their checklist to include a range of features Vocabulary assessment checklist content create embed	**Programming: Further Coding with Scratch** • Understand how to create a simple script in Scratch • Use decomposition to identify key features • Understand what a variable is • Create a variable Vocabulary broadcast block conditional decomposition game negative numbers parameters program script stage code blocks coordinates features information orientation position	**Programming: Computational Thinking** • Understand that problems can be solved more easily using computational thinking • Understand what the different code blocks do and create a simple game • Understand the terms ‘pattern recognition’ and ‘abstraction’ and how they help to solve a problem • Create a Scratch program which draws a square and at least one other shape

	bcc (blind carbon copy) compose cyberbullying domain email email account email address emoji emotions fake font genuine hacker icons inbox information log in negative language personal information link log out password positive language	loop program repetition code scratch animation code coding application decompose game predict remising code review sprite tinker	storyboard trailer video camera angle cross dissolve fade to black film graphics key events photo recording sound effects time code transition voiceover wipe	audience collaboration contribution design evaluate features hobby hyperlinks inset plan homepage images online progress published review subpage theme record style tab web page	project sprite tinker variables	**Vocabulary** abstraction code decomposition logical reasoning pattern recognition sequence algorithm computational thinking input output script variable
UKS2 Y5 & Y6	Programming: Micro:bits • Clip blocks together and predict what will happen. Make connections with previous programming interfaces	Data Handling: Mars Rover • Identify some of the types of data that the Mars Rover could collect • Explain how the Mars Rover transmits the data back to	Skills Showcase – Mars Rover • Create a pixel picture • Save a JPEG as a bitmap • Explain the 'fetch, decode, execute' cycle in relation to	Computing systems and networks • Explain that codes can be used for a number of different reasons and decode messages	Creating Media: History of a Computer • Explain how to record sounds • Product a simple radio play • Create a document that includes correct	Skills Showcase: Inventing a Product • Evaluate code • Debug programs and make more efficient • Design appropriate housing for their product using CAD software

	they've used e.g. Scratch • Create their own images to make the animation and recognise the different between 'on start' and 'forever' • Identify inputs and outputs used and make predictions about how variables work • Choose appropriate blocks to complete the program and attempt the challenges independently • Break a program down into smaller steps, suggesting • Appropriate blocks and match the algorithm to the program **Vocabulary** algorithm	Earth and the challenges involved in this • Read numbers in binary, up to eight bits • Identify input, processing and output on the Mars Rover • Grasp the concept of binary addition • Relate binary signals (Boolean) to a simple character-based language, ASCII **Vocabulary** 8-bit binary ASCII Boolean communicate CPU addition binary code byte construction data transmission decimal discovery hexadecimal instructions mars rover design distance	real world situations • Create a profile with a safe and suitable username and password • Begin to use 3D design tools **Vocabulary** 3D binary image compression data fetch, decode, execute input memory operating system pixels responsible rom algorithm cad CPU drag and drop id card jpeg online community output ram RGB safe	• Explain how to ensure a password is secure • Create a simple poster with information about Bletchley Park • Explain the importance of historical figures and their contribution towards computer science **Vocabulary** acrostic code caesar cipher cipher combination convince discovery invention password pigpen cipher scrambled secure brute force hacking chip and pin system code contribute date shift cipher	date information and facts about computers • Describe all of the features that we'd expect a computer to have including RAM, ROM, hard drive and processor **Vocabulary** background noise computer file gigabyte hard drive byte devices FX graphics hardware kilobytes memory storage operating system play radio play megabyte mouse overlay processor RAM raspberry pi reverb	• Create an appealing website for their product • Create an edited video of their project • Describe and show how to search for information online **Vocabulary** adapt advert bugs algorithm coding debugging design edit electronic evaluate facts image rights images influence information inputs loops manipulation opinions output photos product program

	app Bluetooth connection debug animation blocks code block create decompose designing device images instructions load desktop download input laptop loop	input internet moon numerical data planet ram scientist output radio signal research sequence		pig Latin present secret technological advancement trial and error	script sound record ROM smartphone sound effects	repetition screenshot search engine selection sequence