Brackenwood Junior School

Computing

Long Term Plan

2022-23

Computing Rationale

Computing at Brackenwood Junior School offers an ambitious, progressive curriculum, which equips our pupils to participate in a rapidly changing world. Every day activities are being increasingly transformed by access to varied and developing technology and Computing ensures that our pupils' 'cultural capital' is being developed in conjunction with it.
Pupils use computing to find, explore, analyse and present information responsibly and creatively. It promotes initiative and independent learning, with pupils being able to make informed judgements about when and where to use different programmes and computing skills to best effect.

A high-quality computing education equips pupils to use deeper thinking and digital skills to understand and change the world. Our computing curriculum has deep links with STEM and although Computing meets both the aims and programme of study of the National Curriculum, children are able to develop their basic computing skills through other subject areas.

The Computing curriculum is divided into three main areas: computer science, digital literacy and information technology.
The core area 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 and coding.
The second area of the curriculum is information technology, which deals with applying computer systems to solve real-world problems. Things that have long been part of Computing in school, such as finding things out, exchanging and sharing information, and reviewing, modifying and evaluating work, remain as important now, for a broad and balanced technological education.

The third is digital literacy, where children are able to express themselves and develop their ideas using computer science and information technology at a level suitable for the future workplace and as active participants in a digital world.

Computing Intent

Computer Science	Digital Literacy	Information Technology
Computational thinking, Programming, Computer Networks	Self-image and identity, Online relationships, Online reputation, Online bullying, Managing online information, Health wellbeing and lifestyle, Privacy and security, Copyright and ownership	Word processing/typing, Data handling, Presentations, web design and ebooks, Animation, Video creation, Photography and Digital art, Augmented reality and virtual reality, Sound

	Year 3	Year 4	Year 5	Year 6
Autumn 1	Connecting Computers	The Internet	Sharing Information	Internet Communication
Autumn 2	Desktop Publishing	Creating media – Photo editing	Vector Drawings	Web page creation
Spring 1	Branching databases	Repetition in Shapes	Selection in physical computing	Variables in games
Spring 2	Stop Frame Animation	Data logging	Flat-file databases	Sensing
Summer 1	Sequencing Sounds	Creating media – Audio editing	Video editing	Introduction to spreadsheets
Summer 2	Events and Actions	Repetition in Games	Selection in quizzes	3D modelling

Computing Implementation

Substantive and disciplinary knowledge in computing

Substantive knowledge

Substantive knowledge in computing is understanding how to use technology, how to be safe and knowing how to program. This is developed through deliberate practice and by children applying their knowledge of how to be computational thinkers.
"Computational thinking is an important life skill, which all pupils now need to develop. It is central to both living in and understanding our digitally enriched world. It is a cognitive process involving logical reasoning by which problems are solved across the whole curriculum and through life in general." (Computing at School, 2015)

Disciplinary knowledge

Disciplinary knowledge in computing is the use and interpretation of substantive knowledge in order to develop original digital content and programs.

Creativity

Computing is an area of the curriculum that has many opportunities for children to demonstrate creativity through developing their own programs, systems and digital content whilst applying their developing computational thinking. Computing has opportunities for natural cross-curricular learning; examples include presenting data in tables, researching in History or writing instructions in English.

Computing Progression Map				
	Year 3	Year 4	Year 5	Year 6
Computer Science	To explain how digital devices function. To identify input and output devices. To recognise how digital devices can change the way we work. To explain how a computer network can be used to share information To explore how digital devices can be connected. To recognise the physical components of a network. To identify that commands have an outcome. To combine sound commands into a sequence. To implement an algorithm as code. To explain the relationship between an event and an action. To create a program to move a sprite in four directions. To adapt a program to a new context. To develop my program by adding features. To identify and fix bugs in a program. To design and create a maze-based challenge.	To describe how networks physically connect to other networks. To recognise how networked devices make up the internet. To outline how websites can be shared via the World Wide Web. To describe how content can be added and accessed on the World Wide Web. To create a code snippet for a given purpose. To create a program in a text-based language. To use and modify a count-controlled loop to produce a given outcome. To explain that a computer can repeatedly call a procedure. To develop the use of count-controlled loops in a different programming environment. To recognise and choose between infinite loops and count-controlled loops. To develop a design that includes two or more loops which run at the same time. To modify an infinite loop in a given program. To design and create a project that includes repetition.	To explain that computers can be connected together to form systems. To recognise the role of computer systems in our lives. To recognise how information is transferred over the internet. To explain how sharing information online lets people in different places work together. To control a simple circuit connected to a computer. To write a program that includes count-controlled loops. To explain that a loop can stop when a condition is met, e.g. number of times. To create a controllable system that includes selection. To explain how selection is used in computer programs. To relate that a conditional statement connects a condition to an outcome. To explain how selection directs the flow of a program. To design, create and evaluate a program which uses selection.	To describe how search engines select results. To explain how search results are ranked. To explain why a variable is used in a program and use variables to improve a game. To design, create and evaluate a project including algorithms, variables and artwork. To create a program to run on a controllable device. To explain that selection can control the flow of a program. To update a variable with a user input. To use a conditional statement to compare a variable to a value. To design and develop a project that uses inputs and outputs on a controllable device.

Digital Literacy	To select copyright-free images to use in a publication	To recognise how the content of the WWW is created by people. To evaluate the consequences of unreliable content. To search for, save and edit image from a copyright-free website. To consider why someone might want to change the composition of an image. To recognise that not all images are real.	To contribute to a shared project online. To evaluate different ways of working together online. To demonstrate the safe use and handling of devices.	To identify how to use a search engine. To recognise why the order of results is important, and to whom. To recognise how we communicate using technology. To evaluate different methods of online communication. To consider the ownership and use of images in web site design To recognise the implications of linking to content owned by other people
Information Technology	To identify the object attributes needed to collect relevant data. To select objects to arrange in a branching database. To group objects using my own yes/no questions To prove my branching database works. To explain that questions need to be ordered carefully to split objects into similarly sized groups. To compare two branching database structures. To use a branching database to answer	To explain that data gathered over time can be used to answer questions. To use a digital device to collect data automatically. To explain that a data logger collects 'data points' from sensors over time. To use data collected over a long duration to find information. To identify the data needed to answer questions. To identify changes that we can make to an image including 'crop'.	To create multiple questions about the same field. To order, sort, and group my data cards. To explain what a 'field' and a 'record' is in a database. To compare paper and computer-based databases. To outline how grouping and then sorting data allows us to answer questions. To explain that tools can be used to select specific data. To select an appropriate chart to visually compare data. To refine a chart by selecting a particular filter.	To identify questions which can be answered using data. To apply an appropriate number format to a cell. To explain that formulas can be used to produce calculated data. To apply formulas to data, including duplicating. To create a spreadsheet to plan an event. To choose suitable ways to present data including tables and graphs. To use a computer to create and manipulate 3D digital objects

	questions. To compare the information shown in a pictogram with a branching database. To recognise that text and layout can be edited. To change font style, size, and colours for a given purpose. To explain what 'page orientation' means. To recognise placeholders and say why they are important. To paste text and images to create a magazine cover. To identify and use different layouts. To recognise animation as a sequence of drawings or photographs. To make a flip book animation To recognise why small changes are needed for each frame. To plan an achievable animation using a storyboard. To use onion skinning to help me make small changes between frames. To add media into an animation such as text or sound.	To change the composition of an image by selecting parts of it. To alter an image using different colours, filters and retouching. To combine parts of images to create new images. To identify the inputs and outputs required to play audio or record sound. To use a digital device to record sound. To listen to and identify features of a podcast. To plan and record my own podcast. To save and edit audio recordings. To show that different types of audio can be combined and played together.	To identify that drawing tools can be used to produce different outcomes. To create a vector drawing by combining shapes. To use zoom tools, alignment grids and resize handles. To change the order of layers in a vector drawing. To duplicate and group objects in a vector drawing. To recognise video as moving pictures, which can include audio. To name, identify and use suitable devices for recording video. To investigate further the features of an effective video, including the use of theme, setting, characters, colour, sound, and dialogue. To store, retrieve, and export my recording to a computer To improve a video by reshooting and editing.	To modify a 3D shape by resizing, changing colour, rotating, positioning and duplicating To create digital 3D objects of an appropriate size To choose which 3D objects I need to construct my model To evaluate and modify my model. To review an existing website and consider its structure. To plan a web page including layout, suggested media To recognise the need to preview pages To outline the need for a navigation path

Computing programmes of study: key stages 1 and 2

National curriculum in England

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.

Attainment targets

By the end of each key stage, pupils are expected to know, apply and understand the matters, skills and processes specified in the relevant programme of study.

Schools are not required by law to teach the example content in [square brackets].

Subject content

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.

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.