

Brackenwood Junior School



Computing Long Term Plan

Yearly Overview

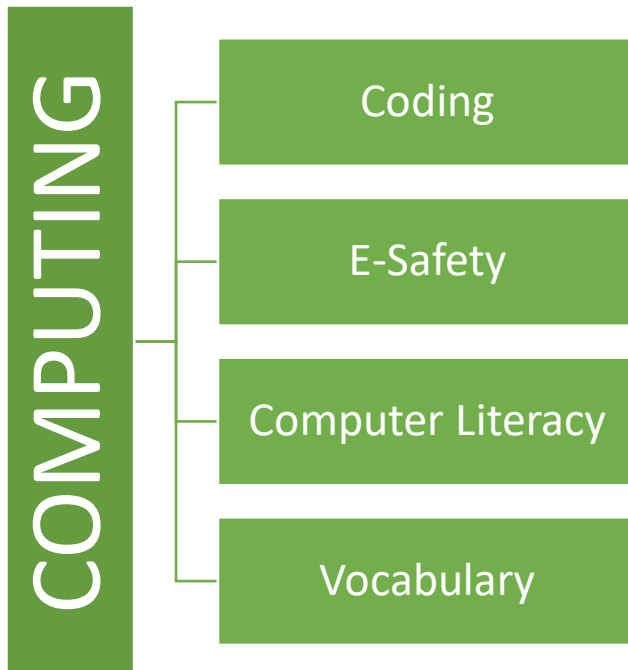
	Autumn	Spring	Summer
Year 3	Coding: Code Studio Course C E-Safety: Kara and Winston Microsoft Word or Google Docs	Coding: Code Studio Course C E-Safety: Kara and Winston Tiny Tap Shadow Puppet Edu Koma Koma	Coding: Code Studio Course C E-Safety: Digizen Preventing Plagiarism Lux Camera Excel/Google Sheets Fact File Databases
Year 4	Coding: Code Studio Course D Commonsense Media: Strong Passwords; Rings of Responsibility Green Screen Microsoft Word or Google Docs	Coding: Code Studio Course D Commonsense Media: This is me, Our Digital Citizenship Pledge Podcasting Decibel X Online Graph Maker	Coding: Code Studio Course D Commonsense Media: The Power of Words, Seeing is Believing Branching Databases IMotion Tiny Tap
Year 5	Coding: Code Studio Course E Commonsense Media: Private and Personal Information; Digital Citizenship Green Screen iMovie PowerPoint	Coding: Code Studio Course E Commonsense Media: My Media Choices; A Creator's Rights and Responsibilities Airtable Excel / Google Sheets	Coding: Code Studio Course E Commonsense Media: Keeping Games Fun and Healthy; Online Tracks Branching Databases Adobe Spark Video Google Earth Pro
Year 6	Coding: Code Studio Course F Commonsense Media: Media Balance; You Won't Believe This Circuit Construction Google Doc / Microsoft Word	Coding: Code Studio Course F Commonsense Media: Beyond Gender Stereotypes, Digital Friends Adobe Spark Page Google Science Journal	Coding: Code Studio Course F Commonsense Media: Cyber Bullying, Reading the News Google Earth Pro Green Screen



Computing

“Computers themselves, and software yet to be developed, will revolutionize the way we learn.” – Steve Jobs

Our key driving themes are:



Why should children learn this subject?	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
What will children learn to do in this subject?	<p>At Brackenwood Junior School, children will:</p> <ul style="list-style-type: none">• Apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation• Analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems• Evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems• Become responsible, competent, confident and creative users of information and communication
How will we inspire them?	<ul style="list-style-type: none">• Provide access to high quality hardware such as ipads and laptops.• Invite computing specialists in to lead workshops• Use coding to create computer games and systems• Link E-safety to their real life experiences

Skills Progression Map

Code Developing an understanding of instructions, logic and sequences	
YEAR 3 AND 4	YEAR 5 AND 6
<ul style="list-style-type: none"> • Use specified screen coordinates to control movement. • Set the appearance of objects and create sequences of changes. • Create and edit sounds. Control when they are heard, their volume, duration and rests. • Control the shade of pens. • Specify conditions to trigger events. • Use IF THEN conditions to control events or objects. • Create conditions for actions by sensing proximity or by waiting for a user input (such as proximity to a specified colour or a line or responses to questions). • Use variables to store a value. • Use the functions define, set, change, show and hide to control the variables. • Use the Reporter operators <p> $() + ()$ $() - ()$ $() * ()$ $() / ()$ to perform calculations. </p>	<ul style="list-style-type: none"> • Set IF conditions for movements. Specify types of rotation giving the number of degrees. • Change the position of objects between screen layers (send to back, bring to front). • Upload sounds from a file and edit them. Add effects such as fade in and out and control their implementation. • Combine the use of pens with movement to create interesting effects. • Set events to control other events by 'broadcasting' information as a trigger. • Use IF THEN ELSE conditions to control events or objects. • Use a range of sensing tools (including proximity, user inputs, loudness and mouse position) to control events or actions. • Use lists to create a set of variables. • Use the Boolean operators $() < ()$, $() = ()$, $() > ()$ $() \text{and} ()$ $() \text{or} ()$ $\text{Not} ()$ to define conditions. • Use the Reporter operators $() + ()$ $() - ()$ $() * ()$ $() / ()$ to perform calculations. Pick Random $()$ to $()$ Join $()$ $()$ Letter $()$ of $()$ Length of $()$ $() \text{ Mod } ()$ This reports the remainder after a division calculation Round $()$ $()$ of $()$.

Connect

Understanding how to safely connect with others

- | | |
|---|---|
| <ul style="list-style-type: none">• Use some of the advanced features of applications and devices in order to communicate ideas, work or messages professionally. | <ul style="list-style-type: none">• Choose the most suitable applications and devices for the purposes of communication.• Use many of the advanced features in order to create high quality, professional or efficient communications. |
|---|---|

Communicate

Using Apps to Communicate Ideas

- | | |
|---|--|
| <ul style="list-style-type: none">• Devise and construct databases using applications designed for this purpose in areas across the curriculum. | <ul style="list-style-type: none">• Select appropriate applications to devise, construct and manipulate data and present it in an effective and professional manner. |
|---|--|

National Curriculum Programmes of Study

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.