COMPUTING

CURRICULUM: COMPUTING

St Bede's
Catholic Primary School

Together with Christ we Grow in Faith and Knowledge

CURRICULUM NARRATIVE

KEY:

- **AL** – Algorithms
- **CS** – Computing Systems
- **CM** – Creating Media
- **DI** – Data & Information
- **DD** – Design & Development
- **ET** – Effective Use of Tools
- **IT** – Impact of Technology
- **NW** – Networks
- **PG** – Programming
- **SS** – Safety & Security

Year 6

- Internet Communication
- Webpage Creation
- Variables In Games
- Intro to Spreadsheets
- 3D Modelling
- Sensing

Welcome to secondary school!

Year 5

- Selection In Quizzes
- Vector Drawing
- Flat-file Databases
- Selection in Physical Computing
- Video Production
- Sharing Information

Year 4

- The Internet
- Audio Production
- Repetition In Shapes
- Data Logging
- Photo Editing
- Repetition In Games

Year 3

- Events & Actions In Programs
- Desktop Publishing
- Branching Databases
- Sequencing Sounds
- Stop-frame Animation
- Connecting Computers

Year 2

- IT Around Us
- Digital Photography
- Robot Algorithms
- Pictograms
- Making Music
- Programming Quizzes

Year 1

- Programming Animations
- Digital Writing
- Grouping Data
- Moving a Robot
- Digital Painting
- Technology Around Us

EYFS

EYFS is mainly centred around play-based, unplugged (no computer) activities that focus on building children's listening skills, curiosity and creativity and problem solving. Technology in the Early Years can mean: taking a photograph with a camera or tablet, searching for information on the internet, playing games on the interactive whiteboard, exploring an old typewriter or other mechanical toys, using a Beebot, watching a video clip or listening to music. Allowing children the opportunity to explore technology in this carefree and often child-led way, means that not only will they develop a familiarity with equipment and vocabulary but they will have a strong start in key stage 1 Computing and all that it demands.

Your Computing journey starts here!

CURRICULUM NARRATIVE

Why do Computer Scientists read?

To find evidence and gather information.

To learn how to perform tasks on computer-based systems

To analyse instructions and make predictions

To evaluate and apply new technologies to existing technologies.

Write like a Computer Scientist

Instruction - Record how to perform a particular function.

Inform – Provide information to impart knowledge.

Critique – Review and analysis of systems and strategies to determine how it is fit for purpose.

Similarity and difference - Analysis of the extent and type of difference between systems and strategies.

Question – Question why and how it does or doesn't work and give possible explanations.

Key Strands

Algorithms – Being able to comprehend, design, create and critique algorithms

Computer networks — Understanding how networks can be used to retrieve and share information, and how they come with associated risks

Computer systems — Understanding what a computer is, and how its constituent parts function together as a whole

Creating media — Selecting and creating a range of media including text, images, sound and video.

Data and information — Understanding how data is stored, organised, and used to represent real-world artefacts and scenarios.

Design and development — Understanding the activities involved in planning, creating, and evaluating computing artefacts.

Effective use of tools — Using software tools to support computing work.

Impact of technology — Understanding how individuals, systems, and society as a whole interact with computer systems

Programming — Creating software to allow computers to solve problems

Safety and security — Understanding risks when using technology, and how to protect individuals and systems

The study of Computing allows pupils to be in control of a technology which is everywhere and will play a pivotal part in their lives. Therefore, we want to model and educate our pupils on how to use technology positively, responsibly and safely. We want our pupils to be creators not consumers and our broad curriculum encompassing computer science, information technology and digital literacy reflects this. We want our pupils to understand that there is always a choice with using technology and as a school we utilise technology to model positive use. We recognise that the best prevention for a lot of issues we currently see with technology/social media is through education. Building our knowledge in this subject will allow pupils to effectively demonstrate their learning through creative use of technology. We recognise that technology can allow pupils to share their learning in creative ways. We also understand the accessibility opportunities technology can provide for our pupils. Our knowledge rich curriculum has to be balanced with the opportunity for pupils to apply their knowledge creatively which will in turn help our pupils become skilful computer scientists.

The units for KS1 and KS2 are based on a spiral curriculum, which means the strands are revisited regularly (at least once in each year group), and pupils revisit each strand through a new unit that consolidates and builds on prior learning within that strand.

CURRICULUM NARRATIVE

Progression Through the Key Strands

Within computing, there is a taxonomy of 10 key strands, which combined, ensure that our pupils can access a deep, interconnected understanding of the subject. Each strand has a combination of skills and concepts that feature throughout the national curriculum. Six of these strands describe broad areas of study within computing. The other four strands describe cross-cutting concepts and skills that are taught and revisited through the first six. Pupils make progress in computing by developing their specific, factual content for the topic (this knowledge is often described as 'substantive knowledge') and the skill development which allows them to gain and develop substantive knowledge (often described as 'disciplinary knowledge'). The threshold concepts relate to different aspects of disciplinary knowledge, and substantive knowledge is vital to all of them.

All learning objectives have been mapped to the National Centre for Computing Education's taxonomy of ten strands, which ensure that units build on each other from one key stage to the next. Within the curriculum, every year group learns through units within the same four themes, which combine the ten strands of the National Centre for Computing Education taxonomy. Content from the NCCE is built upon a taxonomy which helps organise the breadth of the subject of computing. This categorisation consists of ten strands that span the current National Curriculum for computing in England. Each strand has a combination of skills and concepts that feature throughout the National Curriculum. Six of these strands describe broad areas of study within computing. The other four strands describe cross-cutting concepts and skills that are taught and revisited through the first six.

Primary Themes	Computing Systems and Network	Algorithms and Programming	Data and Information	Creating Media
Taxonomy Strands	Computer systems Computer networks	Programming Algorithms Design and development	Data and information	Creating media Design and development
	Effective use of tools			
	Impact of technology			
	Safety and security			

CURRICULUM NARRATIVE

Progression Within a Unit – Learning Graphs

Learning graphs are provided as part of each unit and demonstrate progression through concepts and skills. To learn some of these concepts and skills, pupils need prior knowledge of others, so the learning graphs show which concepts and skills need to be taught first and which could be taught at a different time.

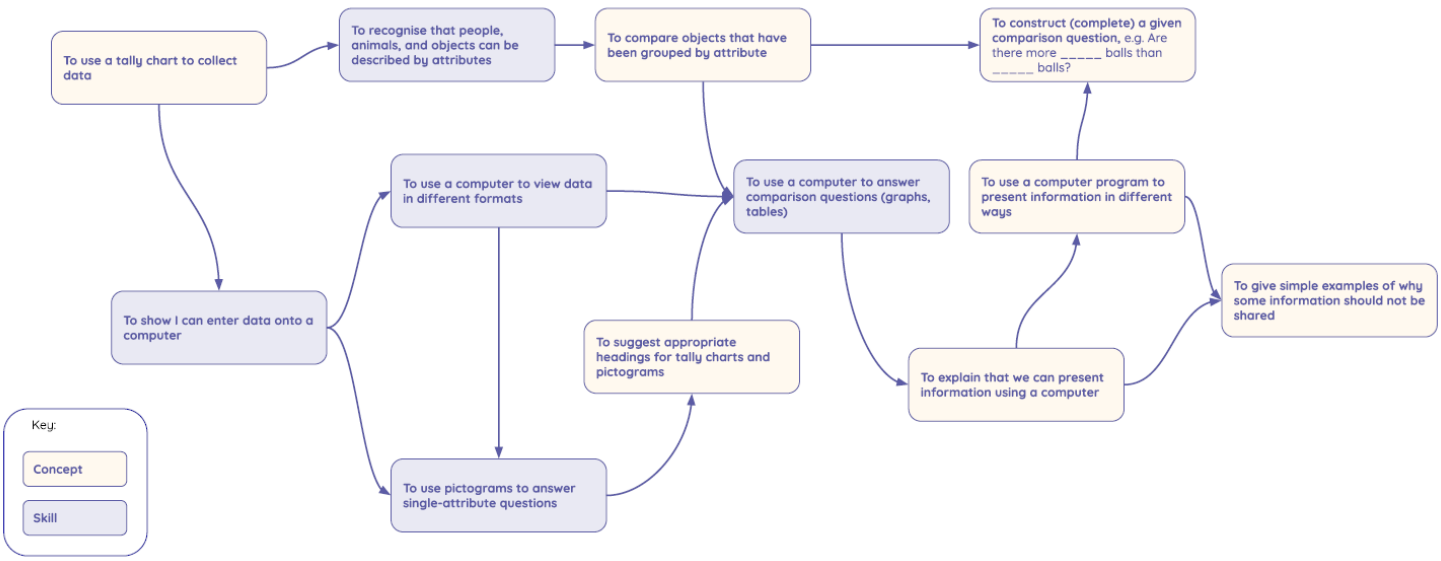
The learning graphs often show more statements than there are learning objectives. All skills and concepts learned are included in the learning graphs. Some of these skills and concepts are milestones, which form learning objectives, while others are smaller steps towards these milestones, which form success criteria.

*Please note:
The wording of the statements may be different in the learning graphs than in the lessons, as the learning graphs are designed for teachers, whereas the learning objectives and success criteria are age-appropriate so that they can be understood by pupils.

In each year group, there are two 'Programming' units of work, but only one 'Programming' learning graphs. The second 'Programming' unit builds on the content that was taught in the first 'Programming' unit so closely that there is no specific divide where one ends and the other begins.

KS1 Example Learning Graph

Year 2 – Data an Information - Pictograms



COMMON THREADS

The curriculum has been developed with key threads underpinning the different units. Effective pedagogy is at the heart of good teaching and learning; successful computing teachers combine their knowledge of the subject with evidence-based teaching practices

Lead with Concepts
Support pupils in the acquisition of knowledge through use of key concepts, terms, and vocabulary, providing opportunities to build a shared and consistent understanding. Glossaries, concept maps, and displays, along with regular recall and revision, can support this approach.

Work Together
Encourage collaboration, specifically using pair programming and peer instruction, and also structured group tasks. Working together stimulates classroom dialogue, articulation of concepts, and development of shared understanding

Get Hands-on
Use physical computing and making activities that offer tactile and sensory experiences to enhance learning. Combining electronics and programming with arts and crafts (especially through exploratory projects) provides pupils with a creative, engaging context to explore and apply computing concepts.

Unplug, Unpack, Repack
Teach new concepts by first unpacking complex terms and ideas, exploring these ideas in unplugged and familiar contexts, then repacking this new understanding into the original concept. This approach (semantic waves) can help pupils develop a secure understanding of complex concepts

Modelling
Model processes or practices — everything from debugging code to binary number conversions — using techniques such as worked examples and live coding. Modelling is particularly beneficial to novices, providing scaffolding that can be gradually taken away.

Program Comprehension
Use a variety of activities to consolidate knowledge and understanding of the function and structure of programs, including debugging, tracing, and Parson's Problems. Regular comprehension activities will help secure understanding and build connections with new knowledge.

Create Projects
Use project-based learning activities to provide pupils with the opportunity to apply and consolidate their knowledge and understanding. Design is an important, often overlooked aspect of computing. Pupils can consider how to develop an artefact for a particular user or function and evaluate it against a set of criteria.

Add Variety
Provide activities with different levels of direction, scaffolding, and support that promote learning, ranging from highly structured to more exploratory tasks. Adapting your instruction to suit different objectives will help keep all pupils engaged and encourage greater independence.

Challenge Misconceptions
Use formative questioning to uncover misconceptions and adapt teaching to address them as they occur. Awareness of common misconceptions alongside discussion, concept mapping, peer instruction, or simple quizzes can help identify areas of confusion.

Contextual Examples
Bring abstract concepts to life with real-world, contextual examples and a focus on interdependencies with other curriculum subjects. This can be achieved through use of unplugged activities, proposing analogies, storytelling around concepts, and finding examples of the concepts in pupils' lives.

Structure Lessons
Use supportive frameworks when planning lessons, such as PRIMM (Predict, Run, Investigate, Modify, Make) and Use-Modify-Create. These frameworks are based on research and ensure that differentiation can be built in at various stages of the lesson

Read and Explore Code First
When teaching programming, focus first on code 'reading' activities, before code writing. With both block-based and text-based programming, encourage pupils to review and interpret blocks of code. Research has shown that being able to read, trace, and explain code augments pupils' ability to write code.

CURRICULUM SUMMARY – KS1 / KS2

	Technology in our lives	Multimedia	Programming	Handling Data
EYFS	Understanding of the world Speaking People, Culture and Communities	Writing Being imaginative and expressive The Natural world/Speaking Creating with materials	Self confidence and self awareness Building relationships Communication and language Managing Self / Self regulation Numerical patterns Speaking	Speaking Building relationships Past and present Communication and language Select and use technology Number Managing self

	Computing systems and network	Creating Media	Programming A	Data and Information	Creating Media	Programming B
Year One	**Technology around us** Recognising technology in school and using it responsibly	**Digital painting** Choosing appropriate tools in a program to create art and making comparisons with working non-digitally.	**Moving a robot** Writing short algorithms programs for floor robots and predicting program outcomes.	**Grouping data** Exploring object labels, then using them to sort and group objects by properties.	**Digital writing** Using a computer to create and format text before comparing to writing non-digitally.	**Programming animations** Designing and programming the movement of a character on screen to tell stories.
Year Two	**Information technology around us** Identifying IT and how its responsible use improves our world in school and beyond	**Digital photography** Capturing and changing digital photographs for different purposes.	**Robot algorithms** Creating and debugging programs and using logical reasoning to make predictions.	**Pictograms** Collecting data in tally charts and using attributes to organise and present data on a computer.	**Digital music** Using a computer as a tool to explore rhythms and melodies, before creating a musical composition.	**Programming quizzes** Designing algorithms and programs that use events to trigger sequences of code to make an interactive quiz.

	Computing systems and network	Creating Media	Programming A	Data and Information	Creating Media	Programming B
Year Three	**Connecting computers** Identifying that digital devices have inputs, processes, and outputs and how devices can be connected to make networks.	**Stop-frame animation** Capturing and editing digital still images to produce a stop-frame animation that will tell a story.	**Sequencing sounds** Creating sequences in a block-based programming language to make music.	**Branching databases** Building and using a branching database to group objects using yes/no questions.	**Desktop Publishing** Creating documents by modifying text, images, and page layouts for a specified purpose.	**Events and actions in programs** Writing algorithms and programs that use a range of events to trigger sequences of actions.
Year Four	**The internet** Recognising the internet as a network of networks including the WWW, and why we should evaluate online content.	**Audio Production** Capturing and editing audio to produce a podcast, ensuring that copyright is considered.	**Repetition in shapes** Using a text-based programming language to explore count-controlled loops when drawing shapes.	**Data logging** Recognising how and why data is collected over time, before using data loggers to carry out an investigation	**Photo editing** Manipulating digital images and reflecting on the impact of changes and whether the required purpose is fulfilled.	**Repetition in games** Using a block-based programming language to explore count-controlled and infinite loops when creating a game.

	Computing systems and network	Creating Media	Programming A	Data and Information	Creating Media	Programming B
Year Five	**Systems and searching** Recognising IT systems in the world and how some can enable searching on the internet.	**Video production** Planning, capturing and editing video to produce a short film.	**Selection in physical computing** Exploring conditions and selection using a programmable microcontroller.	**Flat-file databases** Using a database to order data and create charts to answer questions.	**Introduction to vector graphics** Creating images in a drawing program by using layers and groups of objects.	**Selection in quizzes** Exploring selection in programming to design and code an interactive quiz.
Year Six	**Communication and collaboration** Exploring how data is transferred by working collaboratively online.	**Webpage creation** Designing and creating webpages, giving consideration to copyright, aesthetics, and navigation.	**Variables in games** Exploring variables when designing and coding a game.	**Introduction to spreadsheets** Answering questions by using spreadsheets to organize and calculate data.	**3D Modelling** Planning, developing and evaluating 3D computer models of physical objects.	**Sensing movement** Designing and coding a project that captures inputs from a physical device.

CURRICULUM NARRATIVE

Intent
In Computing we intend to teach the principles of information and computation, how digital systems work, and how to put this knowledge to use through programming. We will build on this knowledge and understanding so that pupils use information technology to create programs, systems and a range of content. We will focus on being safe whilst working in a digital environment and understand the digital footprint we leave. The curriculum will develop pupil's digital literacy so that they are able to use and express themselves at a level suitable for the future workplace and as active participants in a digital world.

Implementation
Our curriculum is based on the National Centre for Computing Education (NCCE) computing taxonomy, which comprises ten key strands of knowledge and skills for Key Stages 1-4, listed below (the strands relevant to KS1 and KS2 are in **bold**).

Each strand summarises the top-level learning outcomes for every unit associated with that strand. For example, the NCCE strand of **'Creating media'** explains that pupils should be able *to 'Select and create a range of media including text, images, sounds, and video'*. All units and lessons within this strand are designed to help children achieve this outcome.

The units for KS1 and KS2 are based on a spiral curriculum, which means the strands are revisited regularly (at least once in each year group), and pupils revisit each strand through a new unit that consolidates and builds on prior learning within that strand.

The Ten NCCE Computing Strands, and Associated Learning Outcomes:

1. Algorithms — Be able to comprehend, design, create, and evaluate algorithms;
2. **Computer networks** — Understand how networks can be used to retrieve and share information, and how they come with associated risks;
3. **Computer systems**— Understand what a computer is, and how its constituent parts function together as a whole;
4. **Creating media** — Select and create a range of media including text, images, sounds, and video;
5. **Data and information** — Understand how data is stored, organised, and used to represent real-world artefacts and scenarios;
6. Design and development — Understand the activities involved in planning, creating, and evaluating computing artefacts;
7. Effective use of tools — Use software tools to support computing work;
8. Impact of technology — Understand how individuals, systems, and society as a whole interact with computer systems;
9. **Programming** — Create software to allow computers to solve problems;
10. Safety and security — Understand risks when using technology, and how to protect individuals and systems.
In Upper KS2, physical computing (using micro-computers such as the Crumble and the micro:bit) is incorporated into the curriculum as it plays an important role in modern pedagogical approaches in computing, both as a tool to engage pupils and as a strategy to develop pupils' understanding in more creative ways. Additionally, physical computing supports and engages pupils in tangible and challenging tasks.

Impact
We encourage our children to enjoy and value the curriculum we deliver. We will constantly ask the WHY behind their learning and not just the HOW. We want learners to discuss, reflect and appreciate the impact computing has on their learning, development and well being. Finding the right balance with technology is key to an effective education and a healthy life-style. We feel the way we implement computing helps children realise the need for the right balance and one they can continue to build on in their next stage of education and beyond. We encourage regular discussions between staff and pupils to best embed and understand this. The way pupils showcase, share, celebrate, and publish their work will best show the impact of our curriculum. We also look for evidence through reviewing pupil's knowledge and skills digitally through online tools and observing learning regularly. Progress of our computing curriculum is demonstrated through outcomes and the record of coverage in the process of achieving these outcomes.

Mixed Age Classes

For mixed age classes, cycles of learning (rolling programmes) ensure that pupils meet key concepts for their year group without repeating the same theme of learning.

CURRICULUM SUMMARY - EYFS

'Digital technology is driving extraordinary global changes, so it's crucial that children and young people are educated to make use of their opportunities.'
- Amanda Spielman

EYFS COVERAGE

Technology in Our Lives
- I can tell you about technology that is used at home and in school.
- **I can operate simple equipment.**
- **I can use a safe part of the Internet to play and learn.**

Multimedia
- I can move objects on a screen.
- **I can create shapes and text on a screen.**
- **I can use technology to show my learning.**

Understanding the world
Children see adults use simple search engines such as Kiddle and Kidrex to find information. They are given opportunities to explore information sites such as Infant Encyclopedia, CBeebies Topics and Glossopedia.

Listening, Attention and Understanding
Children share conversations with experts and other classes using video chat such as Facetime. The classes could be within their own school or in other schools, nationally and internationally.

Writing
Children have a variety of experiences to type their name or label images using 2Publish or other simple software/apps. They are given opportunities to use a tablet and laptop keyboard' and a mouse.

Being imaginative and Expressive, Writing
Children are taught skills to take a photo. They ask permission before taking photos of friends. Children photograph artifacts and scenery that are part of learning. These are added to software and apps for labelling

People, Culture and Communities
Children help their teacher to make decisions about photos that show their learning experiences to a global audience via school website, a blog or Twitter. They are supported to show their learning to family beyond school.

Speaking
Children offer explanations for why things happen as they use different technologies such as a printer, photocopier, microwave and a range of computing devices such as tablets, laptops and interactive whiteboards.

Being imaginative and Expressive
Children use video to retell and create stories. Green screening is used for children to imagine themselves in different places.

The Natural World / Speaking
Children record sounds on a wellie walk or during exploration of musical instruments. Actions are imagined around the sound when it is played back. Children record phrases to describe feelings and objects.

People, Culture and Communities
Children use play technology and junk models to role play work environments such as vets, builders, shop, hospital etc. They have opportunities to explore old technology such as phones, keyboards, old PCs etc.

Understanding the world
Children experience travelling the world using Google Earth software or app. They see photos and visit 3d buildings. Teachers model safe use of YouTube to view videos of places around the globe.

Creating with materials
Children use 2Paint or other simple software/apps to make marks and to paint a picture.

Gross and Fine Motor Skills
IWB is used to encourage big arm movements. Apps and paint software used to develop fine motor control.

Creating with materials
2Simple Music Toolkit, Music Sparkles or Tap a Tune App used to create music. Simple apps and websites used to create animations eg ABCYA.

ONLINE SAFETY / SAFE USE OF TECHNOLOGY

- I use rules given to me by a trusted adult when I use technology.
- I am kind to my friends
- I use a safe part of the Internet to play and learn.
- I make sure a trusted adult is with me.

- I use a log in to access devices.
- I see information that is put online about me.
- I use devices with other people, talking about what we do I am careful with technology devices.

- I tell you the things that are the same about my friend and me.
- I use apps, games and websites that trusted adults show me.
- I use a device for a limited time.

Based on elim-edtech

CURRICULUM COVERAGE – EYFS and KS1

EYFS COVERAGE CONTINUED

Programming

- **I can make a floor robot move.**
- **I can use simple software to make something happen.**
- I make choices about buttons and icons I press, touch or click.

Handling Data

- I can tell you about different kinds of information such as pictures, video, text and sound.

Self confidence and self awareness Children explore the buttons of a floor robots and remote-control toys. They are guided to discover ways to make the object move. Adults talk about what has been achieved by the children.	**Building relationships** Children have opportunities to build environments for floor robots. They work together to navigate the robot or remote control toy around obstacles.	**Speaking / Building Relationships** Children take photos and video to capture learning. They know where it is stored to go back and reflect on their learning and the learning of their friends. They talk about what they can learn from photos and video online or photos in books.	**Past and Present** Children use QR codes to select information they want to find out. Aurasma can be used to recall information that has been collected.
Building relationships / Communication and language Children follow sets of instructions. They communicate instructions to each other and to supporting adults.	**Numerical Patterns** Children use appropriate language to describe position and distance of floor robots. **Number** Children count steps and movement of floor robots.	**Communication and Language** Children collect sounds to provide evidence in an investigation. They record interviews to collect information from adults or other experts. They know where the information is stored and replay it	**Select and use technology** Children use digital microscopes or Magniscope app or a visualiser to examine objects they have collected. They capture the image to label what they have found out.
Self regulation / speaking / Numerical Patterns Children explore apps such as BeeBot / Kodable to make things happen. They talk about solving problems as they work at the low levels in the apps. Adults support them to be willing to make mistakes and learn from them.	**Managing self** Children explore appropriate software, such as 2Go (Simple level) or Poisson Rouge, to make things happen. They are supported to be willing to make mistakes and to find out what they can make happen.	**Number** Children use software such as 2Count and apps such as DoodleBuddy to count information and to talk about what they have found out. They sort photos of things they have investigated.	**Managing self / speaking** Children are supported to know their own important information. They use talking tins or sound recorder apps to practice their information. Adults rehearse who they can tell this information.

National Curriculum Coverage – Years 1 and 2	1.1 – Technology around us	1.2 – Digital Painting	1.3 – Moving a robot	1.4 – Grouping data	1.5 – Digital Writing	1.6 – Programing animations	2.1 – Information technology around us	2.2 – Digital photography	2.3 – Robot algorithms	2.4 – Pictograms	2.5 – Digital music	2.6 – Programming quizzes
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.	✓			✓	✓		✓	✓	✓	✓		

CURRICULUM COVERAGE

National Curriculum Coverage – Years 3 and 4	3.1 Connecting Computing	3.2 Stop-frame animation	3.3 Sequencing sounds	3.4 Branching databases	3.5 Desktop publishing	3.6 – Events and actions in programs	4.1 – The internet	4.2 - Audit production	4.3 – Repetition in shapes	4.4 – Data Logging	4.5 – Photo editing	4.6 – Repetition in games
Design, write and debug programs that accomplish specific goals, including controlling or stimulating 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 and 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; recognize acceptable / unacceptable behaviour; identify a range of ways to report concerns about content and contact.		✓		✓			✓	✓			✓	

National Curriculum Coverage – Years 5 and 6	5.1 – Systems and searching	5.2 – Video production	5.3 – Selection in physical computing	5.4 – Flatfile databases	5.5 – Introduction to vector graphics	5.6 – Selection in quizzes	6.1 – Communication and collaboration	6.2 – Webpage creation	6.3 – Variables in games	6.4 – Introduction to spreadsheets	6.5 – 3D modeling	6.6 – Sensing improvements
Design, write and debug programs that accomplish specific goals, including controlling or stimulating 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 and 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; recognize acceptable / unacceptable behaviour; identify a range of ways to report concerns about content and contact.	✓	✓						✓	✓		✓	

CURRICULUM CONTINUITY – EYFS TO KS1

Based on the National Centre for Computing Education scheme of work we have produced high quality units, following the threshold concepts. An effective computing curriculum must cover all of these concepts across , at least three of these concepts should be covered. Writers of these units have worked to identify sufficient breadth of content and ensure that pupils learn in sufficient depth. The units are written for Year 1 pupils up to Year 6. This document captures the progression from EYFS into Key Stage One and gives suggested activities that could be explored with Early Years pupils to support the computing concepts.

What are the EYFS Computing Skills?

Data and Information	Creating Media	Programming A	Computer systems and networks
• I can tell you about different kinds of information such as pictures, video, text and sound.	• I can move objects on a screen. • I can create shapes and text on a screen. • I can use technology to show my learning.	• make a floor robot move. • use simple software to make something happen. • make choices about buttons and icons I press, touch or click.	• tell you about technology that is used at home and in school. • operate simple equipment. • use a safe part of the Internet to play and learn.

What are the Key Stage 1 Computing Skills?

Data and Information	Creating Media	Programming A	Computer systems and networks	Programming B
• label objects • identify that objects can be counted • describe objects in different ways • count objects with the same properties • compare groups of objects • answer questions about groups of objects • recognise that we can count and compare objects using tally charts • recognise that objects can be represented as pictures • create a pictogram • select objects by attribute and make comparisons • recognise that people can be described by attributes • explain that we can present information using a computer	• describe what different freehand tools do • use the shape tool and the line tools • make careful choices when painting a digital picture • explain why I chose the tools I used • use a computer on my own to paint a picture • compare painting a picture on a computer and on paper • use a computer to write • add and remove text on a computer • identify that the look of text can be changed on a computer • make careful choices when changing text • explain why I used the tools that I chose • compare typing on a computer to writing on paper • use a digital device to take a photograph • make choices when taking a photograph • describe what makes a good photograph • decide how photographs can be improved • use tools to change an image • recognise that photos can be changed • say how music can make us feel • identify that there are patterns in music • experiment with sound using a computer • use a computer to create a musical pattern • create music for a purpose • review and refine our computer work	• explain what a given command will do • act out a given word • combine forwards and backwards commands to make a sequence • combine four direction commands to make sequences • plan a simple program • find more than one solution to a problem • describe a series of instructions as a sequence • explain what happens when we change the order of instructions • use logical reasoning to predict the outcome of a program • explain that programming projects can have code and artwork • design an algorithm • create and debug a program that I have written	• identify technology • identify a computer and its main parts • use a mouse in different ways • use a keyboard to type on a computer • use the keyboard to edit text • create rules for using technology responsibly • recognise the uses and features of information technology • identify the uses of information technology in the school • identify information technology beyond school • explain how information technology helps us • explain how to use information technology safely • recognise that choices are made when using information technology	• choose a command for a given purpose • show that a series of commands can be joined together • identify the effect of changing a value • explain that each sprite has its own instructions • design the parts of a project • use my algorithm to create a program • explain that a sequence of commands has a start • explain that a sequence of commands has an outcome • create a program using a given design • change a given design • create a program using my own design • decide how my project can be improved

CURRICULUM CONTINUITY – LKS2

What are the Lower Key Stage Two Computing Skills?

Data and Information	Creating Media	Programming A	Computer systems and networks	Programming B
• create questions with yes/no answers • identify the attributes needed to collect data about an object • create a branching database • explain why it is helpful for a database to be well structured • plan the structure of a branching database • independently create an identification tool • explain that data gathered over time can be used to answer questions • use a digital device to collect data automatically • explain that a data logger collects 'data points' from sensors over time • recognise how a computer can help us analyse data • identify the data needed to answer questions • use data from sensors to answer questions	• explain that animation is a sequence of drawings or photographs • relate animated movement with a sequence of images • plan an animation • identify the need to work consistently and carefully • review and improve an animation • evaluate the impact of adding other media to an animation • recognise how text and images convey information • recognise that text and layout can be edited • choose appropriate page settings • add content to a desktop publishing publication • consider how different layouts can suit different purposes • consider the benefits of desktop publishing • identify that sound can be recorded • explain that audio recordings can be edited • recognise the different parts of creating a podcast project • apply audio editing skills independently • combine audio to enhance my podcast project • evaluate the effective use of audio • explain that the composition of digital images can be changed • explain that colours can be changed in digital images • explain how cloning can be used in photo editing • explain that images can be combined • combine images for a purpose • evaluate how changes can improve an image	• explore a new programming environment • identify that commands have an outcome • explain that a program has a start • recognise that a sequence of commands can have an order • change the appearance of my project • create a project from a task description • identify that accuracy in programming is important • create a program in a text-based language • explain what 'repeat' means • modify a count-controlled loop to produce a given outcome • decompose a task into small steps • create a program that uses count-controlled loops to produce a given outcome	• explain how digital devices function • identify input and output devices • recognise how digital devices can change the way we work • explain how a computer network can be used to share information • explore how digital devices can be connected • recognise the physical components of a network • describe how networks physically connect to other networks • recognise how networked devices make up the internet • outline how websites can be shared via the World Wide Web (WWW) • describe how content can be added and accessed on the World Wide Web (WWW) • recognise how the content of the WWW is created by people • evaluate the consequences of unreliable content	• explain how a sprite moves in an existing project • create a program to move a sprite in four directions • adapt a program to a new context • develop my program by adding features • identify and fix bugs in a program • design and create a maze-based challenge • develop the use of count-controlled loops in a different programming environment • explain that in programming there are infinite loops and count controlled loops • develop a design that includes two or more loops which run at the same time • modify an infinite loop in a given program • design a project that includes repetition • create a project that includes repetition

CURRICULUM CONTINUITY – UKS2

What are the Upper Key Stage Two Computing Skills?				
Data and Information	**Creating Media**	**Programming A**	**Computer systems and networks**	**Programming B**
• use a form to record information • compare paper and computer-based databases • outline how you can answer questions by grouping and then sorting data • explain that tools can be used to select specific data • explain that computer programs can be used to compare data visually • use a real-world database to answer questions • identify that drawing tools can be used to produce different outcomes • create a vector drawing by combining shapes • use tools to achieve a desired effect • recognise that vector drawings consist of layers • group objects to make them easier to work with • apply what I have learned about vector drawings • create a data set in a spreadsheet • build a data set in a spreadsheet • explain that formulas can be used to produce calculated data • apply formulas to data • create a spreadsheet to plan an event • choose suitable ways to present data	• explain what makes a video effective • identify digital devices that can record video • capture video using a range of techniques • create a storyboard • identify that video can be improved through reshooting and editing • consider the impact of the choices made when making and sharing a video • review an existing website and consider its structure • plan the features of a web page • consider the ownership and use of images (copyright) • recognise the need to preview pages • outline the need for a navigation path • recognise the implications of linking to content owned by other people • recognise that you can work in three dimensions on a computer • identify that digital 3D objects can be modified • recognise that objects can be combined in a 3D model • create a 3D model for a given purpose • plan my own 3D model • create my own digital 3D model	• control a simple circuit connected to a computer • write a program that includes count-controlled loops • explain that a loop can stop when a condition is met • explain that a loop can be used to repeatedly check whether a condition has been met • design a physical project that includes selection • create a program that controls a physical computing project • define a 'variable' as something that is changeable • explain why a variable is used in a program • choose how to improve a game by using variables • design a project that builds on a given example • use my design to create a project • evaluate my project	• explain that computers can be connected together to form systems • recognise the role of computer systems in our lives • experiment with search engines • describe how search engines select results • explain how search results are ranked • recognise why the order of results is important, and to whom • explain the importance of internet addresses • recognise how data is transferred across the internet • explain how sharing information online can help people to work together • evaluate different ways of working together online • recognise how we communicate using technology • evaluate different methods of online communication	• explain how selection is used in computer programs • relate that a conditional statement connects a condition to an outcome • explain how selection directs the flow of a program • design a program which uses selection • create a program which uses selection • evaluate my program • create a program to run on a controllable device • explain that selection can control the flow of a program • update a variable with a user input • use a conditional statement to compare a variable to a value • design a project that uses inputs and outputs on a controllable device • develop a program to use inputs and outputs on a controllable device

SEND

The BHCET Computing curriculum has been designed to be delivered to the whole class. However, the tasks are adapted by class teachers to meet the needs of individual children. To ensure pupils with SEND achieve well, they should be exposed to the same learning as their peers; however, the way they evidence their learning through the tasks can be adapted.

Through scaffolding, tasks can be adapted to ensure all learners can access and evidence the same threshold concepts and learning objectives as their non-SEND counterparts. Scaffolding strategies can include providing sentence starters, a writing frame, vocabulary banks, sorting and matching cards or visual prompts. Reactive or proactive adaptations can make the BHCET curriculum accessible and achievable for all.

Other strategies of adaptation are outlined through the EEF's Five-a-Day principles, which include explicit instruction, metacognitive strategies, flexible grouping and the use of technology:

Scaffolding
'Scaffolding' is a metaphor for temporary support that is removed when it is no longer required. Initially, a teacher would provide enough support so that pupils can successfully complete tasks that they could not do independently. This requires effective assessment to gain a precise understanding of the pupil's current capabilities.

Examples: Support could be visual, verbal, or written. Writing frames, partially completed examples, knowledge organisers, sentence starters can all be useful. Reminders of what equipment is needed for each lesson and classroom routines can be useful. Scaffolding discussion of texts: promoting prediction, questioning, clarification and summarising.

Explicit Instruction
Explicit instruction refers to a range of teacher-led approaches, focused on teacher demonstration followed by guided practice and independent practice. Explicit instruction is not just "teaching by telling" or "transmission teaching" One popular approach to explicit instruction is Rosenshine's 'Principles of Instruction'.

Examples: Worked examples with the teacher modelling self-regulation and thought processes is helpful. A teacher might teach a pupil a strategy for summarising a paragraph by initially 'thinking aloud' while identifying the topic of the paragraph to model this process to the pupil. They would then give the pupil the opportunity to practise this skill. Using visual aids and concrete examples promotes discussion and links in learning.

Cognitive and Metacognitive Strategies
Cognitive strategies are skills like memorisation techniques or subject specific strategies like methods to solve problems in maths. Metacognitive strategies help pupils plan, monitor and evaluate their learning

Examples: Chunking the task will support pupils with SEND – this may be through provision of checklists, instructions on a whiteboard or providing one question at a time. This helps reduce distractions to avoid overloading working memory.
Prompt sheets that help pupils to evaluate their progress, with ideas for further support.

Flexible Grouping
Flexible grouping describes when pupils are allocated to smaller groups based on the individual needs that they currently share with other pupils. Such groups can be formed for an explicit purpose and disbanded when that purpose is met

Examples: Allocating temporary groups can allow teachers to set up opportunities for collaborative learning, for example to read and analyse source texts, complete graphic organisers, independently carry out a skill, remember a fact, or understand a concept. Pre-teaching key vocabulary, is a useful technique.

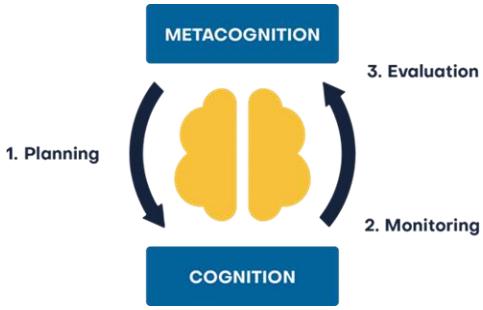
Use of Technology
Technology can assist teacher modelling. Technology, as a method to provide feedback to pupils and/ or parents can be effective, especially when the pupil can act on this feedback.

Examples: Use a visualizer to model worked examples. Technology applications, such as online quizzes can prove effective. Speech generating apps to enable note-taking and extended writing can be helpful.

ASSESSMENT

Assessment comprises two linked processes:

Formative Assessment: provides Assessment **for** Learning. Is a continuous process and an integral part of teaching and learning; informal observations, dialogue/effective use of questioning, consolidation activities, low stakes quizzing, routine marking; and pupil/peer assessment all contribute to the developing profile of progress. When pupils make changes and consider actions to their work, based on the activity, they are 'self-regulating' their work. Self-regulating activities can be termed Assessment **as** Learning. Self-regulated learners are aware of their strengths and weaknesses, and can motivate themselves to engage in, and improve, their learning. Pupils start by **planning** how to undertake a task, working on it while **monitoring** the strategy to check progress, then **evaluating** the overall success.



Metacognitive Regulation Cycle
(EEF Metacognition & Self regulation Guidance)

Summative Assessment: provides Assessment **of** Learning and is a judgement of attainment at key points throughout the year- using past knowledge to measure attainment and progress. Examples of this are standardised tests, tasks and end of term/annual assessments which include a sample of pupil's prior learning.

Assessment is a continuous process which is integral to teaching and learning and:

- Enables an informed judgement to be made about a pupil's understanding, skills, attitude to learning and successful acquisition of knowledge as they move through the curriculum.
- Incorporates a wide range of assessment techniques to be used in different contexts/purposes.
- Is accompanied by **clear assessment criteria** that enables effective marking and feedback, a reliable progress evaluation to be given and demonstrates clearly what a pupil must do to improve.
- Provides feedback recognising achievement, increasing pupil confidence/motivation.
- Supports learning by making clear to pupils: what they are trying to achieve; what they have achieved; what the learning gaps and misconceptions are and what the next steps in learning are.
- Allows regular subject specific extended writing and access to high quality text/ reading.
- Should be moderated and standardised to ensure **purposeful, meaningful, and timely feedback.**
- Includes feedback to pupils to help them understand what they need to improve, challenging them to achieve their target rather than a grade.
- Allows leaders and staff to make timely adaptations to the curriculum.