Computing: Key Knowledge and Skills Progression Overview

Computing Curriculum Intent

At Bracken Leas Primary School, we aim to prepare our learners for their future by giving them the opportunities to gain knowledge and develop skills that will equip them to become independent users and creators of technology in an ever-changing digital world. We want to model and educate our pupils on how to use technology positively, responsibly and safely, valuing the diversity of users and content created. Knowledge and understanding of Information Communication Technology is of increasing importance for our children's futures, both at home and for employment.

Our Computing curriculum focuses on a progression of skills within the strands of Digital Literacy, Computer Science and Information Technology. Online safety is built into the planning throughout the year, across the school to ensure that children become competent in safely using and understanding a range of technology. Our intention is for Computing to support children's creativity and cross-curricular learning, engaging children and enriching their experiences in school.

Computer Science	Children learn to use and express themselves and develop their ideas through ICT for example writing and presenting as well as exploring art and design using multimedia.
Information Technology	Children understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation. Also, to analyse problems to computational terms, and have repeated practical experience in writing computer programs to solve such problems.
Digital Literacy	Children develop practical skills in the safe use of ICT and the ability to apply these skills to solving relevant, worthwhile problems for example understanding the safe use of the internet, networks and email. As a school, we regularly revisit e-safety and its importance.

	1	2	3	4	5	6
EYFS	Provision provides opportunities for children to access BeeBots in order to develop directional (maths link) and programming skills (beginning to develop logical reasoning skills to predict simple outcomes), an interactive screen is used to enable children to develop online and software based navigational skills and to create and manipulate simple digital content, sound buttons are used to record and retrieve digital sound content.					
Year 1	Online Safety Exploring Purple Mash	Lego builders Grouping and sorting	Pictograms Technology outside of school Safer Internet Day Assembly	Maze explorers Spreadsheets	Coding using Scratch Junior	Animated story books Online safety
Year 2	Online safety Touch typing Making music	Creating pictures Online safety	Coding Safer Internet Day Assembly	Online safety Questioning	Spreadsheets Effective searching	Presenting ideas
Year 3	Online safety Touch typing Simulations	Branching databases Graphing	Coding Online safety Safer Internet Day Assembly	Email	Online safety Presenting	Spreadsheets Online safety
Year 4	Online safety Writing for different audiences	Online safety Animation Effective searching	Logo Hardware investigators Safer Internet Day Assembly	Online safety Making music	Online safety Scratch questions and quizzes unit	Spreadsheets
Year 5	Online safety Word processing Touch typing	Word processing Databases	Online safety Game creator Safer Internet Day Assembly	Modelling Coding	Online safety Coding Concept maps	Spreadsheets
Year 6	Touch typing Quizzing	Kodu game making	Blogging Touch typing Safer Internet Day	Online safety Spreadsheets Touch typing	Spreadsheets Networks	Online safety Text adventures Touch typing

	YEAR 1	YEAR 2	YEAR 3	YEAR 4	YEAR 5	YEAR 6
Computer Science	-Understand that an algorithm is a set of instructions used to solve a problem or achieve an objective. -Know that a computer program turns an algorithm into code that the computer can understand. -Be able to work out what is wrong with a simple algorithm when the steps are out of order, e.g. The Wrong Sandwich in Purple Mash and can write their own simple algorithm, e.g. Colouring in a Bird activity. -Know that an unexpected outcome is due to the code they have created and can make logical attempts to fix the code, e.g. Bubbles activity in 2Code. -When looking at a program, children can read code one line at a time and make good attempts to envision the bigger picture of the overall effect of the program. Children can, for example, interpret where the turtle in 2Go	-Explain that an algorithm is a set of instructions to complete a task. When designing simple programs, children show an awareness of the need to be precise with their algorithms so that they can be successfully converted into code. -Create a simple program that achieves a specific purpose. -Identify and correct some errors, e.g. Debug Challenges: Chimp. -Program designs display a growing awareness of the need for logical, programmable steps. -Identify the parts of a program that respond to specific events and initiate specific actions. For example, they can write a cause and effect sentence of what will happen in a program	-Turn a simple real-life situation into an algorithm for a program by deconstructing it into manageable parts. The children's design shows that they are thinking of the desired task and how this translates into code. -Identify an error within their program that prevents it following the desired algorithm and then fix it. -Demonstrate the ability to design and code a program that follows a simple sequence. -Experiment with timers to achieve repetition effects in their programs. -Begin to understand the difference in the effect of using a timer command rather than a repeat command when creating repetition effects. -Children's designs for their programs show that they are thinking of the structure of a program in logical, achievable steps and absorbing some new knowledge of coding structures. For example,	-When turning a real-life situation into an algorithm, the children's design shows that they are thinking of the required task and how to accomplish this in code using coding structures for selection and repetition. -Make more intuitive attempts to debug their own programs. -Use of timers to achieve repetition effects are becoming more logical and are integrated into their program designs. -Understand 'IF statements' for selection and attempt to combine these with other coding structures including variables to achieve the effects that they design in their programs. As well as understanding how variables can be used to store information while a program is executing, children are able to use and manipulate the value of variables. -Use user inputs and outputs such as 'print to screen'. e.g. 2Code.	-Attempt to turn more complex real-life situations into algorithms for a program by deconstructing it into manageable parts. -Test and debug programs and can use logical methods to identify the approximate cause of any bug but may need some support identifying the specific line of code. -Translate algorithms that include sequence, selection and repetition into code with increasing ease and the children's own designs show that they are thinking of how to accomplish the set task in code utilising such structures. Children combine sequence, selection and repetition with other coding structures to achieve their algorithm design. -When children code, they are beginning to think about their code structure in terms of the ability to debug and interpret the code later, e.g. the use of tabs to organise code and the	-Turn a more complex programming task into an algorithm by identifying the important aspects of the task (abstraction) and then decomposing them in a logical way using their knowledge of possible coding structures and applying skills from previous programs. -Test and debug programs as they go and use logical methods to identify the cause of bugs, demonstrating a systematic approach to try to identify a particular line of code causing a problem. -Translate algorithms that include sequence, selection and repetition into code and the children's own designs show that they are thinking of how to accomplish the set task in code utilising such structures, including nesting structures within each other. -Have an improving understanding of variables in coding, outputs such as sound and movement, inputs

	challenges will end up at the end of the program.		repetition and use of timers. -Attempt to 'step through' more complex code in order to identify errors in algorithms and can correct this. e.g. In programs such as Logo, children can 'read' programs with several steps and predict the outcome accurately. -List a range of ways that the Internet can be used to provide different methods of communication. Children can use some of these methods of communication, e.g. being able to open, respond to and attach files to emails using 2Email. Children can describe appropriate email conventions when communicating in this way.	-Program designs show that children are thinking of the structure of a program in logical, achievable steps and absorbing some new knowledge of coding structures. For example, 'IF' statements, repetition and variables. Children can trace code and use step-through methods to identify errors in code and make logical attempts to correct this. In programs such as Logo, children can 'read' programs with several steps and predict the outcome accurately. -Recognise the main component parts of hardware which allow computers to join and form a network. Children's ability to understand the online safety implications associated with the ways the internet can be used to provide different methods of communication is improving.	naming of variables -Understand the value of computer networks but are also aware of the main dangers. -Recognise what personal information is and can explain how this can be kept safe. -Select the most appropriate form of online communications contingent on audience and digital content, e.g. 2Blog, 2Email, Display Boards.	from the user of the program such as button clicks and the value of functions. -Interpret a program in parts and can make logical attempts to put the separate parts of a complex algorithm together to explain the program as a whole. -Understand and explain in some depth the difference between the internet and the World Wide Web. -Know what a WAN and LAN are and can describe how they access the internet in school.
Information Technology	-Sort, collate, edit and store simple digital content e.g. children can name, save and retrieve their work and follow simple instructions to	-Demonstrate an ability to organise data using, for example, a database such as 2Investigate and can retrieve specific data for conducting simple	-Carry out simple searches to retrieve digital content. Children understand that to do this, they are connecting to the internet and using	-Understand the function, features and layout of a search engine. Appraise selected webpages for credibility	-Search with greater complexity for digital content when using a search engine. -Explain in some detail	-Readily apply filters when searching for digital content. -Explain in detail how credible a webpage is and

	access online resources, use Purple Mash 2Quiz example (sorting shapes), 2Code design mode (manipulating backgrounds) or using pictogram software such as 2Count.	searches. -Edit more complex digital data such as music compositions within 2Sequence. -Children are confident when creating, naming, saving and retrieving content. -Use a range of media in their digital content including photos, text and sound.	a search engine such as Purple Mash search or internet-wide search engines. Collect, analyse, evaluate and present data and information using a selection of software, e.g. using a branching database (2Question), using software such as 2Graph. -Consider what software is most appropriate for a given task. -Create purposeful content to attach to emails, e.g. 2Respond.	and information at a basic level. -Make improvements to digital solutions based on feedback. -Make informed software choices when presenting information and data. -Create linked content using a range of software such as 2Connect and 2Publish+. -Share digital content within their community, i.e. using Virtual Display Boards.	how credible a webpage is and the information it contains. -Make appropriate improvements to digital solutions based on feedback received and can confidently comment on the success of the solution. e.g. creating their own program to meet a design brief using 2Code. -Objectively review solutions from others. -Collaboratively create content and solutions using digital features within software such as collaborative mode. -Use several ways of sharing digital content, i.e.2Blog, Display Boards and 2Email.	the information it contains. -Compare a range of digital content sources and are able to rate them in terms of content quality and accuracy. -Use critical thinking skills in everyday use of online communication. -Make clear connections to the audience when designing and creating digital content. -Design and create blogs to become a content creator on the Internet, e.g. 2Blog. -Use criteria to evaluate the quality of digital solutions and are able to identify improvements, making some refinements.
Digital Literacy	-Understand what is meant by technology and identify a variety of examples both in and out of school. -Make a distinction between objects that use modern technology and those that do not e.g. a microwave vs. a chair. -Understand the	-Effectively retrieve relevant, purposeful digital content using a search engine. -Apply learning of effective searching beyond the classroom. Be able to share this knowledge, e.g. 2Publish example template. -Make links between	-Demonstrate the importance of having a secure password and not sharing this with anyone else. -Explain the negative implications of failure to keep passwords safe and secure. -Understand the importance of staying	-Explore key concepts relating to online safety using concept mapping such as 2Connect. -Help others to understand the importance of online safety. -Know a range of ways of reporting inappropriate content and contact.	-Have a secure knowledge of common online safety rules and can apply this by demonstrating the safe and respectful use of a few different technologies and online services. -Implicitly relate appropriate online behaviour to their right to	-Demonstrate the safe and respectful use of a range of different technologies and online services. -Identify more discreet inappropriate behaviours through developing critical thinking, e.g. 2Respond activities. Recognise the value in

	importance of keeping information, such as their usernames and passwords, private and actively demonstrate this in lessons. -Take ownership of their work and save this in their own private space such as their My Work folder on Purple Mash.	technology they see around them, coding and multimedia work they do in school e.g. animations, interactive code and programs. -Know the implications of inappropriate online searches. -Begin to understand how things are shared electronically such as posting work to the Purple Mash display board. -Develop an understanding of using email safely by using 2Respond activities on Purple Mash and know ways of reporting inappropriate behaviours and content to a trusted adult.	safe and the importance of their conduct when using familiar communication tools such as 2Email in Purple Mash. -Know more than one way to report unacceptable content and contact.		personal privacy and mental wellbeing of themselves and others.	preserving their privacy when online for their own and other people's safety.
Key Vocabulary	In addition to the vocabulary for EYFS: Avatar Device Home screen Icon Login Logout Menu internet Password Private Folder	In addition to the vocabulary for Year 1: Implement Interval Interaction Predict Properties Scale Filter Email Personal information Private information Secure	In addition to the vocabulary for Years 1 and 2: Degrees Flowchart Interval Nest Repeat Appropriate Inappropriate Blog Reliable source Spoof	In addition to the vocabulary for Years 1,2 and 3: Coordinates 'if' and 'if/else' conditional statements Prompt Repeat until Variable Citation Cookies Copyright Data analysis	In addition to the vocabulary for Years 1,2,3 and 4: Abstraction Concatenation Decomposition Physical system Encrypt PEGI ratings function Identity theft Ownership Reference	In addition to the vocabulary for Years 1,2,3,4 and 5: Launch command X and y properties Location sharing Expense Percentage Probability Approval Archive QR code DNS (domain name

	Tool bar Data Pictogram Algorithm Code Debugging Sequence Command Animation Copy Edit Click Event Execute Output Programmer Software Cell Clipart Column Row Spreadsheet value	Sharing Block graph Cut Image value Binary tree Database Record Search Browser Digital footprint Domain Search engine Network URL Web address Web page Web site World wide web Fill Rotate e-book Node Presentation	Verify Vlogs Cell address Pie chart Table Attachment BCC and CC Compose Trusted contact Analysis Evaluation Modelling Simulation Solution Axis Tally chart Audio Sound effect Theme Timing Transition Wordart	Malware Phishing Plagiarism Ransomware Spam Virus Budget Calculations Format cell Formula Formula wizard Line graph Format Procedure (Logo) FPS (frames per second) Frame Onion skinning Stop-motion Results page Components CPU Graphics card Hardware Hard drive Motherboard Network card RAM	CAD 2D and 3D printing Attributing Creative commons Text formatting Hyperlink Reliability Responsibility Computational model Totalling tool Field	server) Ethernet Hosting IP address ISP (internet service provider) LAN (local area network) WAN (wide area network) Router Binary Bit Integer Microprocessor Nibble, Byte, Kilobyte, Megabyte, Gigabyte and Terabyte Nanotechnology Transistor Auto fit Categories ribbon Conditional formatting