



St. Bede's Catholic Primary School Curriculum Progression for Computing

	Reception	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Computer Science	EYFS Curriculum	National Curriculum. Pupils should be taught to: <ul style="list-style-type: none"> 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 		National Curriculum. Pupils should be taught to: <ul style="list-style-type: none"> 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 			
Programming	<p>3-4 Yr olds: Understanding the world: explore how things work</p> <p>Reception: Personal, Social and Emotional Development: show resilience and perseverance in the face of a challenge.</p> <p>ELG: Personal, Social and Emotional Development: be confident to try new activities and show independence, resilience and perseverance in the face of a challenge</p>	<ul style="list-style-type: none"> Through use of programmable toys (Bee-Bots) to give basic commands, including forward and backward a set number of steps. Programme a sequence of instructions and observe what happens. Give a set of instructions to complete a basic task, for example to make the Bee Bot follow a route. 	<ul style="list-style-type: none"> Use logical thinking to explore software and make predictions about it's behaviour, eg look under the hood at a Scratch Jr project and explore what the code does. Write an algorithm to create a basic animation using Scratch Jr. Learn what a 'loop' is. Use loops within an animation. 	<ul style="list-style-type: none"> Use logical thinking to explore more complex software and predict what it does. Incorporate loops to make code more efficient. Make amendments to existing code, ie changing the movement of a sprite in a Scratch game. Take a more systematic approach to debugging; identify the error and how 	<ul style="list-style-type: none"> Code simple games with variables (i.e scoring, lives) Use abstraction and pattern recognition to modify code. Understand websites can be altered by amending code. Program an animation. Debug code with support. 	<ul style="list-style-type: none"> Use flowcharts for unplugged tasks to understand how an algorithm can deviate. Use flowchart diagrams to plan and program a quiz. Begin to use nested loops (loops within loops). Debug own code. Write a code to control an external device. Use repetition within a code. 	<ul style="list-style-type: none"> Debugging quickly and effectively to make a program more efficient (debugging challenges). Remix existing code to solve a problem. Changing and adapting programs to personalise them. Evaluating code to understand it's purpose and predict it's application. Introduce the programming



	<p><i>Can be achieved by supporting children to play with control technology toys, such as an electronic keyboard.</i></p> <p><i>Encourage children to speculate on the reasons why something happens or how something works.</i></p> <p><i>Teach and encourage children to click on different icons to cause things to happen in a computer program.</i></p>	<ul style="list-style-type: none"> Identify improvements to their sequence of commands - debugging Above should all be practiced in unplugged sessions as well, i.e. following instructions for a playground game. 		<p>it can be corrected. This to be done through unplugged and software tasks.</p> <ul style="list-style-type: none"> Code a simple game (i.e Slug game). Debug code with support. 		<ul style="list-style-type: none"> Amend a code within a live scenario. 	<p>language Python.</p>
<p>Hardware</p>	<p><i>Access to programmable toys and ICT equipment- talking buttons, microphones, bee bots).</i></p> <p><i>Familiarise with turning on a computer and using a mouse.</i></p>	<ul style="list-style-type: none"> Explore a range of hardware to discover how it works. Familiarising with location of keys on a keyboard. Learn to use a camera to take photos. Understand that computers have inputs (eg keyboard) and outputs (eg monitor) and identify examples. 	<ul style="list-style-type: none"> Understand what a computer is and that it is made of different components. Recognise that technology follows instructions, driven by human input. Learning that we know if technology is doing what we want it to through it's output. Use cameras with greater control. Develop confidence with the use of a keyboard, including touch typing. 	<ul style="list-style-type: none"> Understand the different components of a computer and what they do. Compare different types of computers (desktop, laptop, tablet, super computer, smart fridge etc). Learn what a server does. 	<ul style="list-style-type: none"> Learn what a server does. 	<ul style="list-style-type: none"> Learn that external devices can be programmed by a computer (LEGO, Microbit, Raspberry Pi). Learn the difference between ROM and RAM and how RAM affects the processing of data. Introduce the fetch, decode, execute cycle (how a computer sequences the steps it takes). 	<ul style="list-style-type: none"> Learn about the history of computers and interface and how they have developed. Explore some historic hardware. Use knowledge of history of computers to design a computer of the future. Learn how QR codes and RFID work.



Computational Thinking		<ul style="list-style-type: none">• Learn that 'decomposition' means to break into smaller parts.• Use decomposition to solve unplugged challenges – eg playground games.• Develop skills of sequencing in unplugged activities• Use logical reasoning to predict the behaviour of simple programs.• Learn an algorithm is a series of step by step instructions to carry out a task, eg cookery• Follow a basic set of instructions, eg washing hands• Create a set of instructions, i.e. an algorithm.	<ul style="list-style-type: none">• Articulate what decomposition is.• Decompose a game to identify what elements created it (sound, movement, scores etc).• Use decomposition to decompose a story into smaller parts.• Explain what an algorithm is and follow one (unplugged – Jam Sandwich).• Create a clear and precise algorithm.• Learn that computers execute by following precise instructions.• Incorporate loops within algorithms (unplugged).	<ul style="list-style-type: none">• Use decomposition to explore the code behind an animation.• Learn what abstraction is and explore real world and computational examples (a tube map, buttons on a game controller).• Learn there are different levels of abstraction and how they are useful.• Explain the purpose of an algorithm• Form simple algorithms independently using Scratch.	<ul style="list-style-type: none">• Solve unplugged problems by decomposing them into smaller parts.• Use decomposition to understand the purpose of a script of code.• Use abstraction to identify the important parts when solving unplugged and plugged activities.• Create algorithms for a specific purpose.	<ul style="list-style-type: none">• Decompose a program without support.• Decompose a story to be able to plan a program to animate a story.• Predict how software will work based on previous experience.• Write more complex algorithms for a purpose.	<ul style="list-style-type: none">• Decompose a program into an algorithm.• Use past experiences to solve a problem.• Write increasingly complex algorithms for a purpose, including a range of programming languages and output devices.
-------------------------------	--	--	--	---	--	---	---



<p>Networks</p>				<ul style="list-style-type: none"> • Learn what a network is and its purpose. • Identify the key components of a network, both wired and unwired. • Recognise that the internet is a network. • Learn how data is transferred. 	<ul style="list-style-type: none"> • Consolidate understanding of key components of a network. • Understand that websites and videos are files shared from one computer to another, over a network. • Learn about packets of data through unplugged tasks. • Learn vocabulary of data and transmit. • Learn about the range of purposes of computer networks. 	<ul style="list-style-type: none"> • Understand that computers transfer data in binary form and be introduced to simple binary numbers. • Relate binary to simple character language ASC II (though code breaking). • Learn that message can be sent using binary code and read simple messages. Understand how bit patterns represent images and how these can be compressed. 	<ul style="list-style-type: none"> • Explore the multiple services computer networks can provide.
<p>Information Technology</p>	<p>EYFS Curriculum</p>	<p>National Curriculum. Pupils should be taught to:</p> <ul style="list-style-type: none"> • use technology purposefully to create, organise, store, manipulate and retrieve digital content 	<p>National Curriculum Pupils should be taught to:</p> <ul style="list-style-type: none"> • 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 				



<p>Using Software</p>	<p>3-4 Yr olds: Physical Development: match their developing physical skills to tasks and activities in the setting</p> <p>Reception: Physical Development: develop their small motor skills so that they can use a range of tools competently, safely and confidently.</p> <p>Expressive arts and design: Explore, use and refine a variety of</p>	<ul style="list-style-type: none"> • Take and edit photos using 2Simple programs. • Use paint programs to achieve different effects. • Develop control of the mouse through dragging and resizing of objects and images. 	<ul style="list-style-type: none"> • Develop word processing skills, including changing font and size, use of underline, copying and pasting text. • Use software to create story animations. 	<ul style="list-style-type: none"> • Use photographs and video to record and tell a story. • Use software to edit video 	<ul style="list-style-type: none"> • Use editing software for video to add transitions, text onscreen and sound. • Use Powerpoint/Prezi, to create presentations • Work collaboratively with others. • Create spreadsheets to represent, sort and analyse data. • Create charts with spreadsheet data. 	<ul style="list-style-type: none"> • Use music software, such as Sonic Pi, to create digital music. • Use animation software and/or video editing tools to create a short stop motion animation. • Identify ways to improve final products. • Introduce 3D design software Tinker CAD, to enable product design. 	<ul style="list-style-type: none"> • Use logical thinking to explore software independently and work towards a collaborative project and final goal using experience of a range of software introduced in previous years.
<p>Using Email and the Internet</p>	<p>artistic effects to express their ideas and feelings.</p> <p>ELG: Personal, Social and Emotional Development: explain the reasons for rules, know right from wrong and try to behave accordingly</p> <p>Expressive arts and design:</p>	<ul style="list-style-type: none"> • Search for and download images from internet safely. 		<ul style="list-style-type: none"> • Learn to log in and out of an email account – Google classroom. • Write an email and send, including addressee. • Attach a document to email. • Reply to an email. 		<ul style="list-style-type: none"> • Develop search skills to find relevant information on the internet. • Learn to use specific search queries and evaluating useful search results. 	<ul style="list-style-type: none"> • Understand that search engines are huge databases, and explore how search engines work, including how they sort and rank results. Use unplugged activities to model.



<p>Using Data</p>	<p>Safely use and explore a variety of materials, tools and techniques, experimenting with colour, design, texture, form and function.</p> <p><i>Can be achieved by being given opportunities to use ICT equipment (computers, tablets) and shown how to carry out simple operations, such as turning on computer or using a mouse. Access to games to support learning on the computer (phonics, numeracy). Provide supported opportunities to carry out basic research into topics and children's interest, using the internet.</i></p>	<ul style="list-style-type: none"> • Introduce databases and spreadsheets. • Children to use 2Simple programs to sort and explore pre-made databases to find information. 	<ul style="list-style-type: none"> • Collect and input own data into simple spreadsheets and databases. • Create charts, pictograms, using data. 	<ul style="list-style-type: none"> • Understand vocabulary related to databases: sort, field, record, data • Learning about the benefits and restrictions of digital and paper databases. • Introduce use of Excel and Access. • Sort and filter databases to find information. • Create charts and graphs using databases. 	<ul style="list-style-type: none"> • Carry out a project to gather data for a purpose (travel survey, weather station etc) and use Excel to represent the data. 	<ul style="list-style-type: none"> • Use some simple formulas (add, total) and sort data within Excel. • Gather and analyse data in real time. 	<ul style="list-style-type: none"> • Understand how data is collected in a range of ways, from surveys to location data on smartphones. • Explore real world examples of data collection (Google Maps, traffic data etc).
<p>Wider Use of Technology</p>	<p><i>Opportunities given to use simple software such as painting programmes to create artwork</i></p>	<ul style="list-style-type: none"> • Identify uses of technology outside of school. 	<ul style="list-style-type: none"> • Learn how computers are used in the wider world, including exploring jobs with computers. 	<ul style="list-style-type: none"> • Understand the purpose and use of emails. 	<ul style="list-style-type: none"> • Understand that software can be used collaboratively as part of team, and use it in such a way. 	<ul style="list-style-type: none"> • Learn what a search engine is. 	<ul style="list-style-type: none"> • Learn about the Internet of Things and how it has led to big data. • Learn how big data can be used to solve problems.



Digital Literacy	EYFS Curriculum	National Curriculum Pupils should be taught to:		National Curriculum Pupils should be taught to:					
	<p>• recognise common uses of information technology beyond school</p> <p>• 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.</p>	<p>• 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</p> <p>• use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.</p>		<p>• 3-4 Yr olds: Personal, Social and Emotional Development: remember rules without needing an adult to remind them</p> <p>Reception: Personal, Social and Emotional Development: know and talk about the different factors that support their health and wellbeing – sensible amounts of screen time</p> <p>ELG: Personal, Social and Emotional Development: explain the reasons for rules, know right from wrong and try to behave accordingly</p> <p><i>Can be achieved by providing role play opportunities where technology may be</i></p>	<ul style="list-style-type: none"> Learn to save and open work into their own folder. Understand the importance of a password and not sharing this with others. Learning what to do if they come across something online that worries or upsets them. 	<ul style="list-style-type: none"> Independently save own work and open work from a saved location. Understand how to stay safe when talking to someone online, who they should and shouldn't be talking to online and what to do if an online experience worries or upsets them. 	<ul style="list-style-type: none"> Understanding their responsibilities when using online services and how to show respect to others. How to be a good digital citizen, recognising online behaviour which is not acceptable, and what to do if they come across this. Learn about cyberbullying. Learn about how not all emails are genuine, how to spot this and what to do about it. 	<ul style="list-style-type: none"> Learning what appropriate behaviour is when working online with someone. Recognising that not all information on the internet is correct or true and how to identify this. 	<ul style="list-style-type: none"> Identify possible dangers online and how to stay safe. Learning to use online communities safely and learning at which age certain online communities are and aren't appropriate.



	<p><i>present (home, office, shop) and toys to support this play. Discussion of how technology can be used safely (ie always with an adults permission). Support children to co-ordinate actions to use technology, for example, call a telephone number.</i></p> <p><i>Have initial discussions around e-safety, such as only using when adults allow, limiting screentime, etc</i></p>						
--	--	--	--	--	--	--	--