



Subject Handbook

Computing



St. John the Baptist
Catholic Primary School



Our Computing Curriculum



Intent



At St John the Baptist Catholic Primary School we recognise that technology is an integral part of every day life.

We aim to prepare our children for a future in an environment which is shaped by technology, by providing them with a high-quality computing education which equips them to use programming, computational thinking and creativity to understand and change the world.

As well as the benefits of Computing, we are also aware of the risks. Online Safety is at the core of our Computing curriculum, we also prepare our children to stay safe online through the use of eCadets, a program which trains, supports and empowers pupils to become online safety and digital citizenship experts.

St. John the Baptist
Catholic Primary School



Our Computing Curriculum



Implementation



At St John the Baptist, Computing is taught on a weekly basis with a new unit of work each half term. Having discreet lessons means that children are able to develop depth in their knowledge and skills throughout the Computing units.

We always start the unit with an Online Safety lesson followed by the unit of work from the NCCE Teach Computing curriculum. In Computing lessons, children have access to a class set of laptops and iPads; these are shared across the school and are available on a timetabled basis. There are further laptops available if needed to ensure that all year groups have the opportunity to use a range of devices and programs for many purposes across the wider curriculum, as well as in discrete computing lessons.

The implementation of the Teach Computing curriculum ensures a balanced coverage of the three strands of the Computing National Curriculum: Computer science, Information technology and Digital literacy. The children will have experiences of all three strands in every year group, ensuring that learning is built upon year on year.

We ensure that staff have the relevant subject knowledge through the use of Teach Computing courses and webinars that are offered from Computing At School. Every teacher has completed the Overview to Computing for their Year group.

Children's work for Computing is collected in a number of ways. Through paper-based evidence and the use of digital evidence folders.



Our Computing Curriculum



Impact



Our curriculum is fun and engaging and relevant to the future. Our Computing lessons will provide pupils with a secure and comprehensive knowledge of technology and digital systems that will ensure that they become confident to use them in their future studies and this digital age. This will enable our pupils to pursue a wide range of interests and vocations in the next stage of their lives.

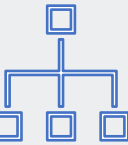
Through our Computing lessons and the eCadet program, we believe our children will be equipped with the skills and knowledge to use technology effectively, and most importantly, they will be able to do so safely. The biggest impact we want is that our children understand the consequences of using the internet and are aware of how to keep themselves safe online.

We measure the impact of our curriculum using the following methods: · In KS1 Summative assessment is carried out through evidence of pupils' work, discussions, photographs and pupil self-assessment. · In KS2 every unit includes an optional summative assessment in the form of a multiple-choice quiz or a rubric. Both are designed to ensure the assessment of pupils' understanding of computing concepts and skills. Formative assessment opportunities are available for teachers to use at the end of each lesson, this allows for any misconceptions to be recognised and addressed.

The Computing subject leader uses teachers' assessment to ensure children are achieving expected standard at the end of each year group. This is done in the following ways:

- Looking at children's work saved onto their individual accounts
- Interviewing the pupils about their learning (pupil voice).
- Class portfolios are scrutinised and there is the opportunity for a dialogue between teachers to understand their class's work.
- Annual reporting of standards across the curriculum.

Curriculum Rationale



Programme Structure

In **Life to the Full**, we follow a four-stage structure which is repeated and developed across four different learning stages:

- **Early Years Foundation Stage** is aimed at Preschool and Reception
- **Key Stage One** is aimed at Years 1 and 2
- **Lower Key Stage Two** is aimed at Years 3 and 4
- **Upper Key Stage Two** is aimed at Years 5 and 6

Within each learning stage, there are three Modules which are based on the Model Catholic RSE Curriculum:

- **Created and Loved by God**
- **Created to Love Others**
- **Created to Live in Community**

Each **Module** is then broken down into **Units** of Work.

Module 1	Created and Loved by God
Units	Religious Understanding Me, My Body, My Health Emotional Well-being Life Cycles
Module 2	Created to Love Others
Units	Religious Understanding Personal Relationships Life Online Keeping Safe
Module 3	Created to Live in Community
Units	Religious Understanding Living in the Wider World

Within each Unit there are **a number of planned sessions** which are to be led in the classroom.

Programme Content

The programme adopts a spiral curriculum approach so that as a child goes through the programme year-after-year, the learning will develop and grow, with each stage building on the last.

Module One: Created and Loved by God

Module One: Created and Loved by God explores the individual. Rooted in the teaching that we are made in the image and likeness of God, it helps children to develop an understanding of the importance of valuing themselves as the basis for personal relationships.

Module Two: Created to Love Others

Module Two: Created to Love Others explores the individual's relationship with others. Building on the understanding that we have been created out of love and for love, this Module explores how we take this calling into our family, friendships and relationships, and teaches strategies for developing healthy relationships and keeping safe both online and in our daily lives.

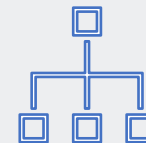
Module Three: Created to Live in Community

Finally, Module Three: Created to Live in Community explores the individual's relationship with the wider world. Here we explore how human beings are relational by nature and are called to love others in the wider community through service, through dialogue and through working for the Common Good.

In the first Unit, Religious Understanding, the sessions help children to develop a concept of the Trinity at a level appropriate for their learning stage.

In subsequent Unit 2 sessions, we apply this religious understanding to real-world situations, such as the community we live in, and through exploring the work of charities which work for the Common Good.

Protected Characteristics Coverage



The Life to the Full programme

How does Life to the Full meet the statutory guidance on Protected Characteristics?

Life to the Full is a programme in Relationships and Health Education for Catholic primary schools, and Relationships, Sex and Health Education for Catholic secondary schools in England, and Relationships and Sexuality Education in Wales, which aims to put the God-given dignity of the human person at the core of the teaching. It is a 14-year programme of study stretching from Early Years to Sixth Form which supports and embraces the statutory guidance whilst providing a Christian vision of the human person, in all of his or her complexity, diversity and richness, rooted in the teaching of the Catholic Church.

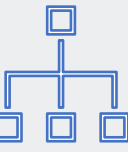
School leaders will be particularly keen to know where and how the programme specifically addresses the expectations raised by Ofsted in the previous section so they can be evidenced during inspection; this section aims to provide some specific guidance.

It should be noted, however, that **Life to the Full** is an evolving programme. We garner, listen to, and respond to feedback by making changes and improvements to ensure that the programme is continually 'alive' and responsive to the needs of pupils, teachers and parents. This document, therefore, may be updated over time, so please do return in the future for updated guidance.

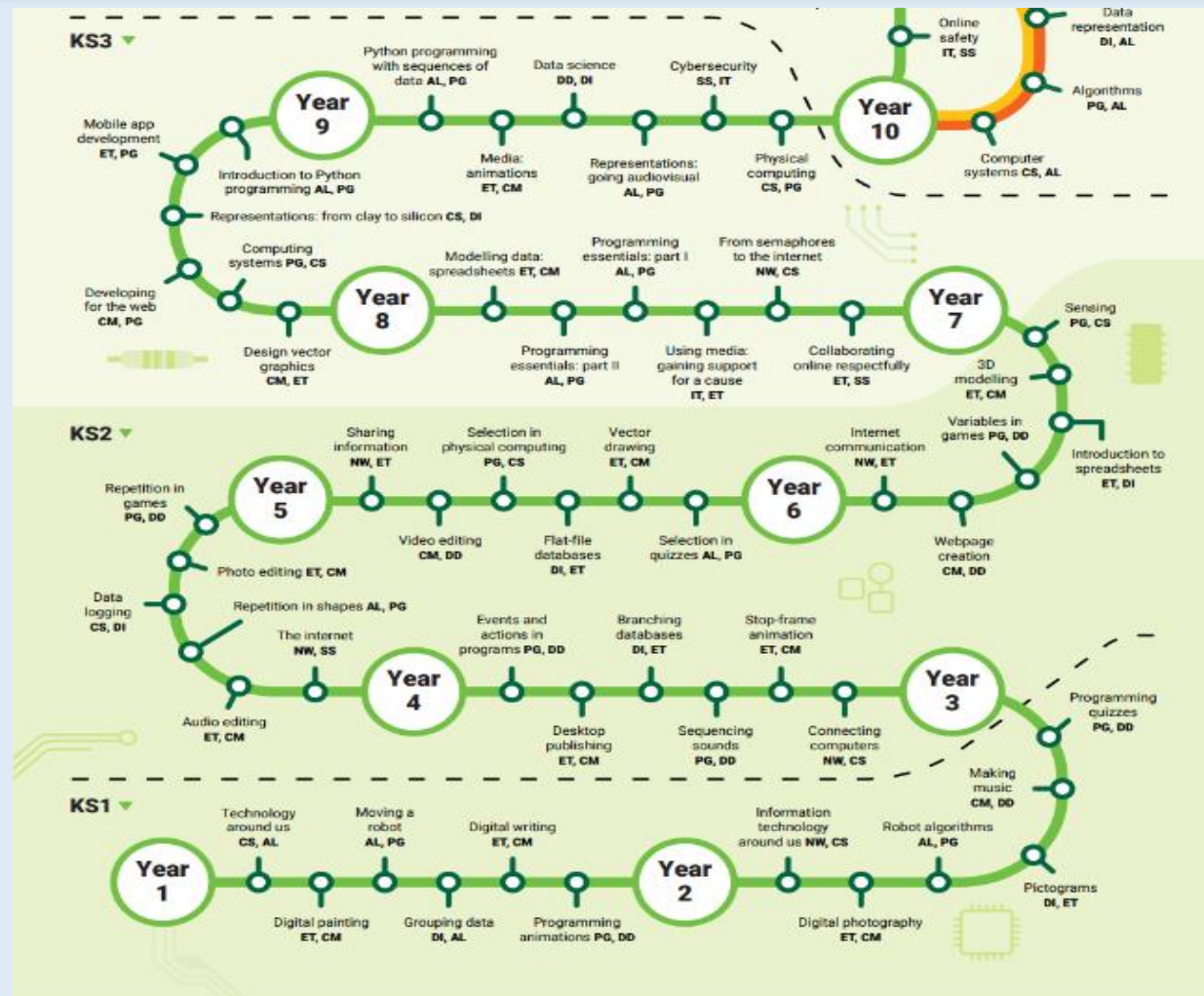
Within the design, structure and presentation of the programme, we have been mindful of the protected characteristics, endeavouring to produce a programme that is inclusive. Our focus has been to inform, present and explain legal rights, underline the fundamental dignity and worth of the person, and outline the erroneous nature of all unjust discrimination. At the same time, in age-appropriate ways, the programme articulates and commends the teaching of the Catholic Church, with special regard to relationships, sex and marriage.



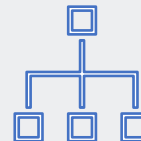
Curriculum Overview



	AUTUMN		SPRING		SUMMER	
	Computing systems and networks	Creating Media A	Programming A	Data and Information	Media B	Programming B
Y1	Technology Around U	Digital Painting	Moving a Robot	Grouping Data	Digital Writing	Programming Animations
Y2	Information Technology Around Us	Digital Photography	Robot Algorithms	Pictograms	Making Music	An Introduction to Quizzes
Y3	Connecting Computers	Stop-frame Animation	Sequence in Music	Branching Databases	Desktop Publishing	Events and Actions
Y4	The Internet	Audio Editing	Repetition in Shapes	Data Logging	Photo Editing	Repetition in Games
Y5	Sharing Information	Video Editing	Selection in Physical Computing	Flat-file Databases	Vector Drawing	Selection in Quizzes
Y6	Communication	Web Page Creation	Variables in Games	Introduction to Spreadsheets	3D Modelling	Sensing



Online Safety Overview



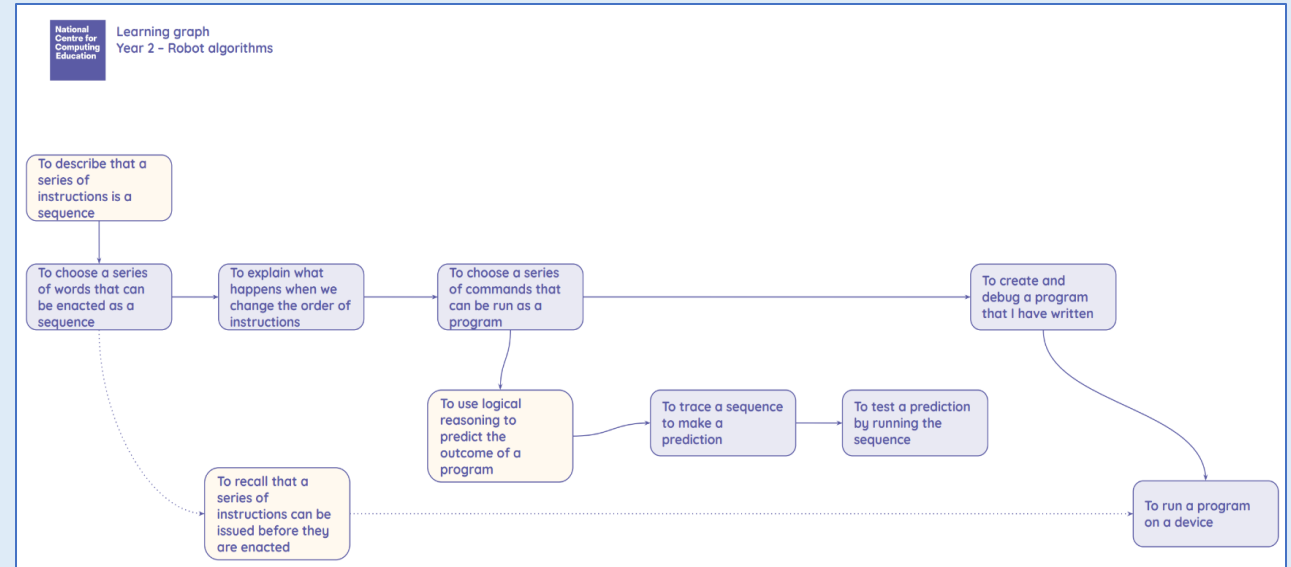
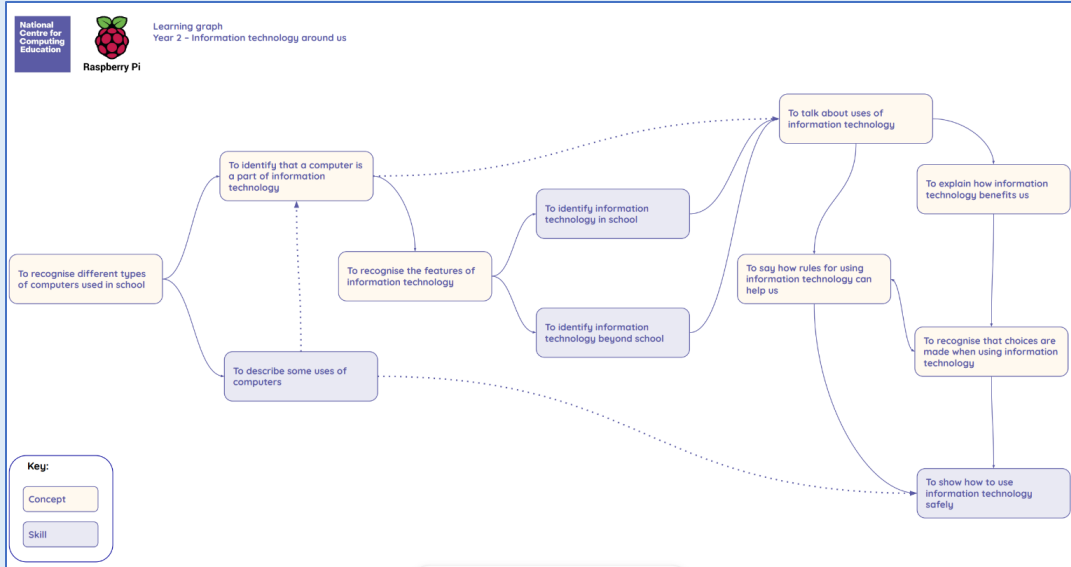
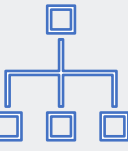
Year 1 Computing	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
	Technology around us Recognising technology in school and using it responsibly. <u>NC Links</u> Use technology purposefully to create, organise, store, manipulate and retrieve digital content Recognise common uses of information 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 <u>Vocabulary</u> Technology Computer Mouse/trackpad Keyboard Screen Click Drag Draw Double-click Input device Shift Space bar Capital letter Full stop	Digital painting Choosing appropriate tools in a program to create art, and making comparisons with working non-digitally. Cross curricular link: Art <u>NC Links</u> Use technology purposefully to create, organise, store, manipulate and retrieve digital content <u>Vocabulary</u> Tool Paintbrush Erase Fill Undo shape tools Line tool Brush style Brush size	Moving a robot Writing short algorithms and programs for floor robots, and predicting program outcomes. <u>NC Links</u> 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 Recognise common uses of information technology beyond school <u>Vocabulary</u> Forwards Backwards Turn Clear Go Commands Instructions Directions Plan Algorithm Program Route	Grouping data Exploring object labels, then using them to sort and group objects by properties. <u>NC Links</u> Use technology purposefully to create, organise, store, manipulate and retrieve digital content Recognise common uses of information technology beyond school <u>Vocabulary</u> Object Label Group Search Image Property Data set Value Less Most Fewest Same	Digital writing Using a computer to create and format text, before comparing to writing non-digitally. Cross Curricular link: English – writing <u>NC Links</u> Use technology purposefully to create, organise, store, manipulate and retrieve digital content 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 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 <u>Vocabulary</u> Word processor Keyboard Keys Letters Microsoft Word Google Docs Numbers Space Backspace Text cursor Capital letters Toolbar Bold Italic Underline Font Undo	Programming animations Designing and programming the movement of a character on screen to tell stories. <u>NC Links</u> 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 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 <u>Vocabulary</u> ScratchJr Bee-Bot CommandSprite Compare Programming Block Joining Start block Run Program Background Delete Reset Algorithm Predict Effect Change Value Instructions

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 1
Year 2	<p>Information technology around us Identifying IT and how its responsible use improves our world in school and beyond.</p> <p><u>NC Links</u> *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</p> <p><u>Vocabulary</u> Information Technology (IT) Computer Barcode scanner/scan</p>	<p>Digital photography Capturing and changing digital photographs for different purposes. <i>Cross Curricular link: Art</i></p> <p><u>NC Links</u> *Use technology purposefully to create, organise, store, manipulate and retrieve digital content *Recognise common uses of information technology beyond school</p> <p><u>Vocabulary</u> Device Camera Photograph Capture Image Digital Landscape Portrait Horizontal Vertical Field of view Narrow Wide Framing Focal point Subject Compose Natural lighting Artificial lighting Flash Focus Background Foreground Editing tools Filter</p>	<p>Robot algorithms Creating and debugging programs, and using logical reasoning to make predictions.</p> <p><u>NC Links</u> *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</p> <p><u>Vocabulary</u> Instructions Sequence Clear Unambiguous Algorithm Program Sequence Order Commands Prediction route debugging</p>	<p>Pictograms Collecting data in tally charts and using attributes to organise and present data on a computer.</p> <p><u>NC Links</u> *Use technology purposefully to create, organise, store, manipulate and retrieve digital content *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><u>Vocabulary</u> Organise Data Object Tally Chart Votes Total Pictogram Compare Count More than Less than Explain Most common Least common Attribute Block diagram</p>	<p>Making music Using a computer as a tool to explore rhythms and melodies, before creating a musical composition.</p> <p><u>NC Links</u> *Use technology purposefully to create, organise, store, manipulate and retrieve digital content</p> <p><u>Vocabulary</u> Music Pattern Rhythm Pulse Tempo Pitch Notes Instrument Open Edit</p>	<p>Programming quizzes Designing algorithms and programs that use events to trigger sequences of code to make an interactive quiz.</p> <p><u>NC Links</u> *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</p> <p><u>Vocabulary</u> Sequence Command Program Run Start Outcome Predict Blocks Sprite Algorithm Design Actions Project Modify Debug</p>

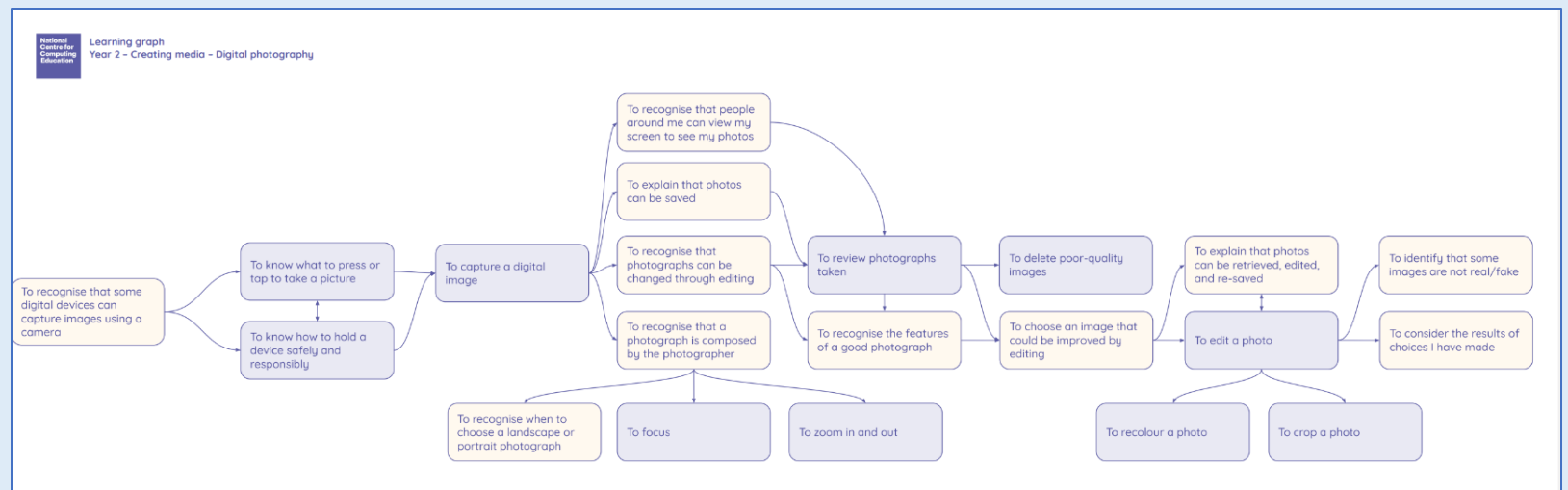
	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 1	
Year 3	Connecting computers Identifying that digital devices have inputs, processes, and outputs, and how devices can be connected to make networks. <u>NC Links</u> *Use sequence, selection, and repetition in programs; work with variables and various forms of input and output * 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 <u>Vocabulary</u> Digital device Input Output Process Program Connection Network Network switch Server Wireless Access Point (WAP)	Stop-frame animation Capturing and editing digital still images to produce a stop-frame animation that tells a story. <u>NC Links</u> * 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 <u>Vocabulary</u> Animation Flip book Stop-frame Sequence Image Photograph Onion-skinning Delete Frame Media Import Transition	Sequencing Sounds Creating sequences in a block-based programming language to make music. <u>NC Links</u> *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 *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 <u>Vocabulary</u> Scratch Programming	Branching Databases Building and using branching databases to group objects using yes/no questions. <u>NC Links</u> * 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 <u>Vocabulary</u> Attribute Value Questions Table Objects Branching database Compare Organise Pictogram Decision tree	Desktop Publishing Creating documents by modifying text, images, and page layouts for a specified purpose. <u>NC Links</u> *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 <u>Vocabulary</u> Text Images Communicate Font Style Template Landscape Portrait Orientation Placeholder Desktop publishing Copy	Events and Actions in Programs Writing algorithms and programs that use a range of events to trigger sequence of actions. <u>NC Links</u> *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 *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 <u>Vocabulary</u> Search engine Google DuckDuckGo Index Crawler Bot Ranking Search engine optimisation Links Web crawlers Content creator	

Year 6	Internet communication	Webpage creation	Variables in games	Introduction to spreadsheets	3D modelling	Sensing
	Recognising how the WWW can be used to communicate and be searched to find information.	Designing and creating webpages, giving consideration to copyright, aesthetics, and navigation.	Exploring variables when designing and coding a game.	Answering questions by using spreadsheets to organise and calculate data.	Planning, developing, and evaluating 3D computer models of physical objects.	Designing and coding a project that captures inputs from a physical device.
	<u>NC Links</u> *Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts *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	<u>NC Links</u> *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	<u>NC Links</u> *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 *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	<u>NC Links</u> *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 <u>Vocabulary</u> Spreadsheet Data Data heading Cells Columns and rows Application Format Common attribute Formula Calculation Input Output Cell reference Range Duplicate	<u>NC Links</u> *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 <u>Vocabulary</u> 3D 3D View Space Rotate Position Select Duplicate Dimensions Modify	<u>NC Links</u> *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 *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 <u>Vocabulary</u> Micro:bit MakeCode Input Process Output USB Condition If then else Variable Random Input Selection Sensing
	<u>Vocabulary</u> Search engine Google DuckDuckGo Index Crawler Bot Ranking Search engine optimisation Links Web crawlers Content creator	<u>Vocabulary</u> Website Web page Browser Media Hypertext Markup Language (HTML) Logo Layout Header Copyright Fair use Home page Device Google Sites Breadcrumb trail Navigation hyperlink	<u>Vocabulary</u> Variable Value Event Algorithm Code Task debug			

Learning Graphs



Year 2



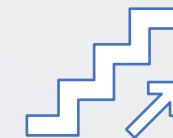
Progression of Vocabulary



Year 1		
Computing systems and networks - Technology around us	Creating media - Digital painting	Creating media - Digital writing
technology, computer, mouse, trackpad, keyboard, screen, double-click, typing.	paint program, tool, paintbrush, erase, fill, undo, shape tools, line tool, fill tool, undo tool, colour, brush style, brush size, pictures, painting, computers	word processor, keyboard, keys, letters, type, numbers, space, backspace, text cursor, capital letters, toolbar, bold, italic, underline, mouse, select, font, undo, redo, format, compare, typing, writing.
Data and information – Grouping	Programming A - Moving a robot	Programming B – Programming animations
object, label, group, search, image, property, colour, size, shape, value, data set, more, less, most, fewest, least, the same	Bee-Bot, forwards, backwards, turn, clear, go, commands, instructions, directions, left, right, route, plan, algorithm, program.	ScratchJr, command, sprite, compare, programming, area, block, joining, start, run, program, background, delete, reset, algorithm, predict, effect, change, value, instructions, design.

Year 2		
Computing systems and networks - Information technology around us	Creating media - Digital music	Creating media - Digital photography
Information technology (IT), computer, barcode, scanner/scan	music, quiet, loud, feelings, emotions, pattern, rhythm, pulse, pitch, tempo, rhythm, notes, create, emotion, beat, instrument, open, edit.	device, camera, photograph, capture, image, digital, landscape, portrait, framing, subject, compose, light sources, flash, focus, background, editing, filter, format, framing, lighting,
Data and information – Pictograms	Programming A - Robot algorithms	Programming B - Programming quizzes
more than, less than, most, least, common, popular, organise, data, object, tally chart, votes, total, pictogram, enter, data, compare, objects, count, explain, attribute, group, same, different, conclusion, block diagram, sharing	instruction, sequence, clear, unambiguous, algorithm, program, order, prediction, artwork, design, route, mat, debugging, decomposition	sequence, command, program, run, start, outcome, predict, blocks, design, actions, sprite, project, modify, change, algorithm, build, match, compare, debug, features, evaluate, decomposition, code.

Progression of Vocabulary



Year 3		
Computing systems and networks – Connecting computers	Creating Media – Desktop publishing	Creating Media – Stop-frame animation
digital device, input, process, output, program, digital, non-digital, connection, network, switch, server, wireless access point, cables, sockets	text, images, advantages, disadvantages, communicate, font, style, landscape, portrait, orientation, placeholder, template, layout, content, desktop publishing, copy, paste, purpose, benefits.	animation, flip book, stop-frame, frame, sequence, image, photograph, setting, character, events, onion skinning, consistency, evaluation, delete, media, import, transition.
Data and Information – Branching databases	Programming A – Sequencing sounds	Programming B – Events and actions in programs
attribute, value, questions, table, objects, branching, database, objects, equal, even, separate, structure, compare, order, organise, selecting, information, decision tree.	Scratch, programming, blocks, commands, code, sprite, costume, stage, backdrop, motion, turn, point in direction, go to, glide, sequence, event, task, design, run the code, order, note, chord, algorithm, bug, debug, code.	motion, event, sprite, algorithm, logic, move, resize, extension block, pen up, set up, pen, design, action, debugging, errors, setup, code, test, debug, actions.

Year 4		
Computing systems and networks – Connecting computers – The internet	Creating Media – Audio production	Creating Media – Photo editing
internet, network, router, security, switch, server, wireless access point (WAP), website, web page, web address, routing, web browser, World Wide Web, content, links, files, use, download, sharing, ownership, permission, information, accurate, honest, content, adverts	audio, microphone, speaker, headphones, input device, output device, sound, podcast, edit, trim, align, layer, import, record, playback, selection, load, save, export, MP3, evaluate, feedback.	image, edit, digital, crop, rotate, undo, save, adjustments, effects, colours, hue, saturation, sepia, vignette, image, retouch, clone, select, combine, made up, real, composite, cut, copy, paste, alter, background, foreground, zoom, undo, font.
Data and Information – Data logging	Programming A – Repetition in shapes	Programming B – Repetition in games
data, table, layout, input device, sensor, logger, logging, data point, interval, analyse, dataset, import, export, logged, collection, review, conclusion.	Logo (programming environment), program, turtle, commands, code snippet, algorithm, design, debug, pattern, repeat, repetition, count-controlled loop, value, trace, decompose, procedure.	Scratch, programming, sprite, blocks, code, loop, repeat, value, infinite loop, count-controlled loop, costume, repetition, forever, animate, event block, duplicate, modify, design, algorithm, debug, refine, evaluate.

Year 5		
Computing systems and networks – systems and searching	Creating Media – Introduction to vector graphics	Creating Media – Video production
system, connection, digital, input, process, storage, output, search, search engine, refine, index, bot, ordering, links, algorithm, search engine optimisation (SEO), web crawler, content creator, selection, ranking.	vector, drawing tools, object, toolbar, vector drawing, move, resize, colour, rotate, duplicate/copy, zoom, select, align, modify, layers, order, copy, paste, group, ungroup, reuse, reflection	video, audio, camera, talking head, panning, close up, video camera, microphone, lens, mid-range, long shot, moving subject, side by side, angle (high, low, normal), static, zoom, pan, tilt, storyboard, filming, review, import, split, trim, clip, edit, reshoot, delete, reorder, export, evaluate, share.
Data and Information – Flat-file databases	Programming A – Selection in physical computing	Programming B – Making Quizzes
database, data, information, record, field, sort, order, group, search, value, criteria, graph, chart, axis, compare, filter, presentation.	microcontroller, USB, components, connection, infinite loop, output component, motor, repetition, count-controlled loop, Crumble controller, switch, LED, Sparkle, crocodile clips, connect, battery box, program, condition, Input, output, selection, action, debug, circuit, power, cell, buzzer	Selection, condition, true, false, count-controlled loop, outcomes, conditional statement, algorithm, program, debug, question, answer, task, design, input, implement, test, run, setup, operator

Year 6		
Computing systems and networks – Communication and collaboration	Creating media – Webpage creation	Creating Media 3D Modelling
communication, protocol, data, address, Internet Protocol (IP), Domain Name Server (DNS), packet, header, data payload, chat, explore, slide deck, reuse, remix, collaboration, internet, public, private, one-way, two-way, one-to-one, one-to-many.	website, web page, browser, media, Hypertext Markup Language (HTML), logo, layout, header, media, purpose, copyright, fair use, home page, preview, evaluate, device, Google Sites, breadcrumb trail, navigation, hyperlink, subpage, evaluate, implication, external link, embed.	TinkerCAD, 2D, 3D, shapes, select, move, perspective, view, handles, resize, lift, lower, recolour, rotate, duplicate, group, cylinder, cube, cuboid, sphere, cone, prism, pyramid, placeholder, hollow, choose, combine, construct, evaluate, modify.
Data and Information – Introduction to spreadsheets	Programming – Variables in games	Programming – Sensing movement
data, collecting, table, structure, spreadsheet, cell, cell reference, data item, format, formula, calculation, spreadsheet, input, output, operation, range, duplicate, sigma, propose, question, data set, organised, chart, evaluate, results, sum, comparison, software, tools.	variable, change, name, value, set, design, event, algorithm, code, task, artwork, program, project, code, test, debug, improve, evaluate, share, assign, declare	Micro:bit, MakeCode, input, process, output, flashing, USB, trace, selection, condition, if then else, variable, random, sensing, accelerometer, value, compass, direction, navigation, design, task, algorithm, step counter, plan, create, code, test, debug.

Assessment

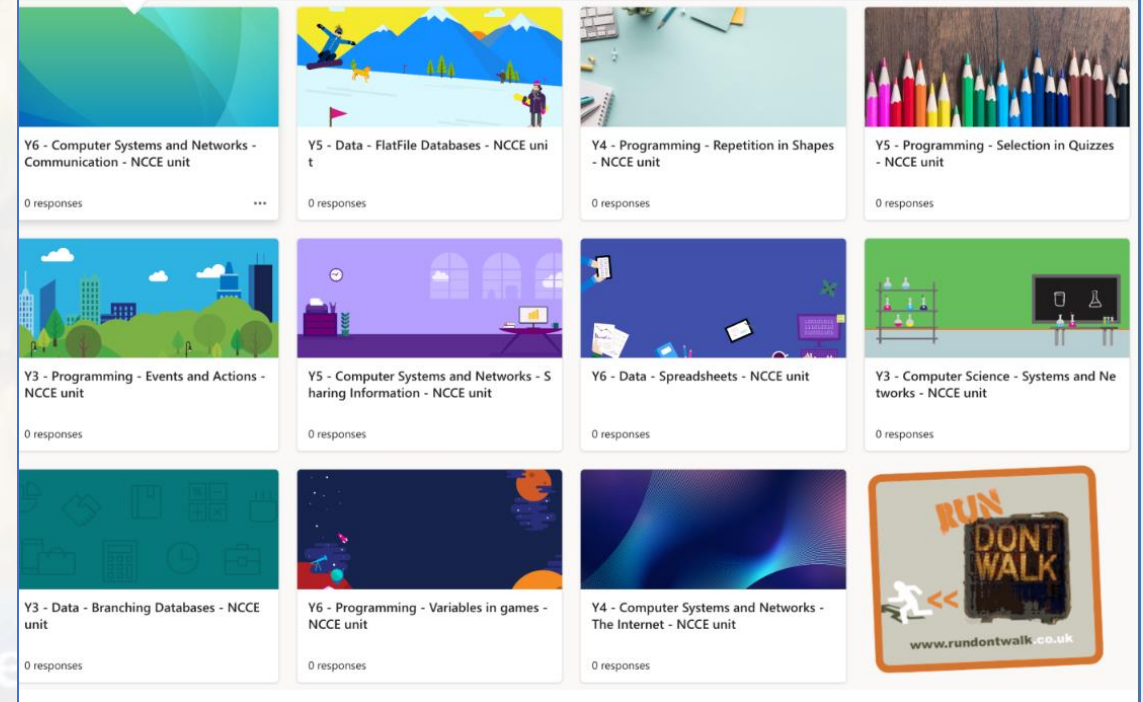


The Computing subject leader uses teachers' assessment to ensure children are achieving expected standard at the end of each year group.

This is done in the following ways:

- Looking at children's work saved onto their individual accounts
- Interviewing the pupils about their learning (pupil voice).
- Class portfolios are scrutinised and there is the opportunity for a dialogue between teachers to understand their class's work.
- Annual reporting of standards across the curriculum.

Assessment quizzes in MS & Google Forms



Inclusion



Accessibility on Microsoft Office/365

If you are using Microsoft Office or Office 365, you have access to the following tools (availability may depend on the version that you are using):

- **Text to speech:** In Microsoft Word, you can add the option to **Speak selected text** to the Quick Access Toolbar (at the top of the page, above the ribbon). Click on the downward arrow on the right-hand side of the Quick Access Toolbar to access the menu of commands.

iPad apps

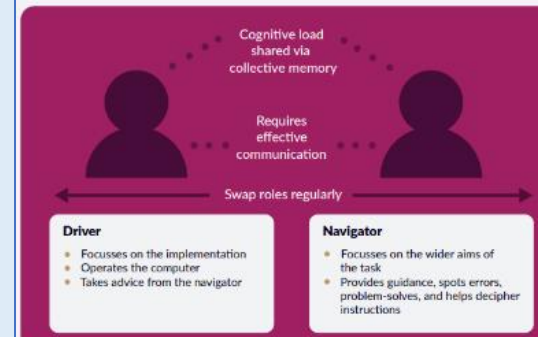
Text-To-Speech – scan text in and it will be read out for the pupil

[ClaroSpeak – Literacy Support](#) - free

[Speechify](#) – free

[Speak Screen](#) – Apple info on accessibility

Pair programming is a pedagogical approach that you can use in your classroom which involves learners working together on a problem to develop programs. This Quick Read aims to highlight the benefits of the approach, as well as factors to consider when applying pair programming in the classroom.



What is pair programming?

Pair programming is an approach where two people work together to write a program or solve a problem whilst sharing a single computer. Pair programming is routinely used in the software industry and soon came to education as the observed benefits became clear.

Application of this concept is more structured than simply asking two learners to work together. Pairing learners without giving guidance as to how you want them to work together can often lead to one, or both, learners quickly losing focus. There needs to be an initial investment of time to develop effective paired work. Ideally, both learners should be **engaged and contributing equally** to the task. Poor communication can be detrimental to the pair's collaboration and can cancel out the benefits of pair programming. Therefore, an essential part of making pair programming a success is spending time ensuring that learners have a good understanding of the roles that they will fulfil during the task.

The **driver** will control the keyboard, mouse, or pen, depending on the task. They will type the code or write out the algorithm as instructed by the navigator. These tasks have a low-level cognitive demand for the learner and allow them to concentrate on writing code accurately, rather than also having to focus on tasks such as problem-solving, deciphering the instructions, and algorithm development.

The **navigator** will support the driver, watching with a keen eye for any errors being made. The navigator will also play a strategic role by thinking of alternative solutions to problems, reading the notes from the teacher, or even walking around the class to look at what others are doing. These tasks have a higher cognitive demand than the tasks of the driver, but as the navigator doesn't have the responsibility of having to write the code, the extraneous load on each member of the pair is reduced.

Summary

Driver/navigator

- Learners take turns playing the role of the driver and the navigator, swapping roles at regular intervals
- The **driver** controls the keyboard and mouse and will write the code
- The **navigator** focuses on the wider aims of the task, spots errors, problem-solves, and reads out instructions to the driver

Benefits

- Reduction in individual **cognitive load** via the collective working memory effect
- Improved confidence in finding solutions, particularly among female students
- Improved quality of programs (fewer errors, more efficient and elegant code)
- Retention of learners' interest in the activities, lessons, and subject

Key considerations

- Communication is key: spend time modelling, emphasising, and rewarding these skills
- Spend time ahead of the lesson carefully planning the pairings based on skills, personalities, or friendships
- Ensure that both the driver and navigator are always working on the same task at the same time
- Experiment with length of intervals to suit your learners' needs
- Ensure that summative assessment is based on paired and individual work/tests, with a greater weighting to individual work
- Check that both members of the pair are fulfilling their roles, and do not allow one to dominate