John Keble CE School

Computing Curriculum

Rooted together in love, growing without limits.

Believing in the worth of every individual, we are a nurturing, Christian sanctuary of learning, where all can flourish. We aspire for everyone to achieve heights of success, to deepen courage and to experience breadth of creativity, knowing the joy of God's love.

Whole school curriculum intent
Our ambitious, knowledge-rich curriculum has been sequenced to equip our pupils with the knowledge and skills to ensure they are happy, healthy global citizens, ready to take their place in modern Britain. The broad and balanced curriculum is creative, coherent and inclusive and, together with our Christian values, enables the pupils to be self-motivated, independent learners.
Subject specific curriculum intent: computing
At John Keble, we desire to give every pupil to understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation. They will be able to analyse problems in computational terms and have repeated practical experience of writing computer programs in order to solve such problems. Pupils will evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems. Finally, pupils will be taught to be responsible, competent, confident and creative users of information and communication technology.
Implementation and impact:
The units in The Teach Computing Curriculum have been written to support all pupils. Each lesson is sequenced so that it builds on the learning from the previous lesson and where appropriate, activities are scaffolded so that all pupils can succeed and thrive. Scaffolded activities provide pupils with extra resources, such as visual prompts, to reach the same learning goals as the rest of the class. Exploratory tasks foster a deeper understanding of a concept, encouraging pupils to apply their learning in different contexts and make connections with other learning experiences. As well as scaffolded activities, embedded within the lessons are a range of pedagogical strategies, which support making computing topics more accessible. The Teach Computing Curriculum uses the National Centre for Computing Education's computing taxonomy to ensure comprehensive coverage of the subject. All learning outcomes can be described through a high-level taxonomy of ten strands: algorithms, computer networks, computer systems, creating media, data and information, design and development, effective use of tools, impact of technology, programming and safety and security. At the end of each unit, pupils will be given an end of unit task. This will be an opportunity for the pupils to showcase their learning and what they have understood in a task. Class teachers will be able to use it as a tool to assess the pupils.

Introduction to John Keble's key stage 1 and 2 computing curriculum

Each unit of work is 6 lessons long and designed to last roughly one-half term. The units for key stages 1 and 2 are based on a spiral curriculum. This means that each of the themes is revisited regularly (at least once in each year group) and pupils revisit each theme through a new unit that consolidates and builds on prior learning within that theme. This style of curriculum design reduces the amount of knowledge lost through forgetting, as topics are revisited yearly. It also ensures that connections are made even if different teachers are teaching the units within a theme in consecutive years.

Year Group	Term	Unit Title	Year Group	Term	Unit Title
1	Autumn 1		2	Autumn 1	
	Autumn 2	Recognising common uses of information technology beyond school		Autumn 2	
	Spring	Algorithms and programming		Spring	Algorithms and programming
	Summer 1			Summer 1	
	Summer 2	Digital media		Summer 2	Using technology purposefully to organise, store, retrieve and manipulate digital content.
3	Autumn 1		4	Autumn 1	
	Autumn 2	Programming with scratch jnr		Autumn 2	Programming quizzes
	Spring 1	Stop-frame animation		Spring 1	Repetition in shapes
	Spring 2	Branching databases		Spring 2	Photo editing
	Summer 1	Connecting computers		Summer 1	Data logging
	Summer 2	Events and actions in programs		Summer 2	Repetition in games
5	Autumn 1		6	Autumn 1	
	Autumn 2	variables in games		Autumn 2	1) Using search technology effectively 2) Computer networks
	Spring 1	Introduction to vector graphics		Spring 1	Webpage creation
	Spring 2	Systems and searching		Spring 2	
	Summer 1	Flat-file databases / Selection in physical computing		Summer 1	
	Summer 2	Flat-file databases / Selection in physical computing		Summer 2	1) Algorithms and programming 2) Sensing movement

National Curriculum – Key Stage 1

National Curriculum objectives	Where covered
● Understand what algorithms are, how they are implemented as programs on digital devices and that programs execute by following precise and unambiguous instructions	Yr 1 Sp / Yr 2 Sp
● Create and debug simple programs	Yr 1 Sp / Yr 2 Sp
● Use logical reasoning to predict the behaviour of simple programs	Yr 1 Sp / Yr 2 Sp
● Use technology purposefully to create, organise, store, manipulate and retrieve digital content	Yr 1 Su2 / Yr 2 Su2
● Recognise common uses of information technology beyond school	Yr 1 A2 / Yr 2 Su2
● 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	Yr 1 Su2 / Yr 2 Su2

National Curriculum – Key Stage 2

National Curriculum objectives	Where covered
• design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts	Yr 3 A2 / Yr 3 Su2 / Yr 4 A2 / Yr 4 Sp 1 / Yr 4 Su2 / Yr 5 A2 / Yr 5 Su2 / Yr 6 Su2
• use sequence, selection and repetition in programs; work with variables and various forms of input and output	Yr 3 A2 / Yr 3 Su1 / Yr 3 Su2 / Yr 4 A2 / Yr 4 Sp 1 / Yr 4 Su1 / Yr 4 Su2 / Yr 5 A2 / Yr 5 Su2 / Yr 6 Su2
• use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs	Yr 3 A2 / Yr 3 Su2 / Yr 4 A2 / Yr 4 Sp 1 / Yr 4 Su2 / Yr 5 A2/ Yr 5 Su2 / Yr 6 Su2
• 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	/ Yr 3 Su1 / Yr 5 Sp2 / Yr 6 A2
• use search technologies effectively, appreciate how results are selected and ranked and be discerning in evaluating digital content	Yr 4 Sp 2 / Yr 5 Su1 / Yr 6 A2 / Yr 6 Sp
• 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	Yr 3 Sp1 / Yr 3 Sp2 / Yr 3 Su2 / Yr 4 A2 / Yr 4 Sp 1 / Yr 4 Sp 2 Yr 4 Su1 / Yr 4 Su2 / Yr 5 A2 / Yr 5 Sp1 / Yr 5 Sp2 / Yr 5 Su1 / Yr 5 Su2 / / Yr 6 A2 / Yr 6 Sp / Yr 6 Su2
• use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.	Yr 3 Sp1 / Yr 3 Sp2/ Yr 4 Sp 2 / Yr 5 A2 / Yr 5 Sp2 / Yr 6 A2 / Yr 6 Sp / Yr 6 Su2

Cross curricular lesson objectives are highlighted in the following colours, alongside any cross curricular vocabulary

Art
English
Maths
DT
History
Music
Science
PSHE

Year 1 – Computing			
Term	**Children learn about…**	**Children learn how to…**	**Vocabulary**
Autumn 2	Recognising common uses of information technology beyond school Downloaded NCCE unit	Develop their understanding of technology and how it can help us. Become familiar with the different components of a computer by developing their keyboard and mouse skills. Use technology responsibly. Links to NCCE curriculum: Computing systems and networks – Technology around us	Backwards, capital letter, clear, click and drag click, commands Computer, Device, directions double-click, drag, draw, Forwards, full stop go, Input, Instructions, keyboard, left, mouse/trackpad, plan, program, responsibly, right, route, safely, screen, shift, space bar, Technology turn,
	Online relationships Online reputation	I can give examples of when I should ask permission to do something online and explain why this is important. I can describe what information I should not put online without asking a trusted adult first.	
Spring 2	Algorithms and programming Downloaded NCCE unit updated Spring 22	Recognise the word algorithm Create an algorithm as a set of instructions to play a simple game like snakes and ladders or how to make a sandwich Program a Beebot forwards, backwards, left and right. As chn build on their skills the sequence of instructions will become longer Identify an error made by a teacher in the programming of the Beebot and correct errors in their own programming Watch a sequence of instructions and predict where the Beebot will go Links to NCCE curriculum: Programming A – Moving a robot	
Summer 2	Digital media Using technology purposefully to create, store and retrieve digital content Downloaded NCCE unit updated Summer 22	Experiment with a simple graphics package e.g Paint to create images and effects with: Lines by changing the size of brushes in response to ideas Shapes using eraser, shape and fill tools Colours and Texture using simple filters to manipulate and create images Use digital technology to save/store and retrieve content. These might include laptop computers, tablets, digital cameras Links to NCCE curriculum: Creating media – Digital painting	
	Copyright and ownership	I understand that work created by others does not belong to me even if I save a copy	

Year 2 – Computing			
Term	**Children learn about…**	**Children learn how to…**	**Vocabulary**
Spring	Algorithms and programming Creating and debugging simple programs *(e.g. program BeeBots/Dash robots to travel around the local area in relation to topic)* Downloaded NCCE unit updated Summer 22	Use the word algorithm to describe a set of instructions they have created Give logical explanations for what they think a program will do and why Identify algorithms in everyday life such as how to cross a road or how to solve an addition / subtraction problem Program a robot to move in a set sequence, e.g. they may program it to move along a map they have created to get from A to B Use an iPad to control Dash robots Correct errors in their code Links to NCCE curriculum: Programming A – Robot algorithms	algorithm, artwork, camera, capture clear, compose debugging, design, digital editing, filter flash, focus, framing, image, instruction, landscape, light sources, lighting, mat order, photograph, portrait prediction, sequence, subject
Summer 2	Using technology purposefully to organise, store, retrieve and manipulate digital content. Downloaded NCCE unit updated Summer 22	Take and save photos, creating folders to store pictures in, and add these to presentations and documents (e.g. take photos of plants over a period of time and then use them to create a video presentation) Retrieve content and edit it, e.g. open a photo or document which has been saved for them and edit List a number of ways IT is used beyond school including for adults to share work and discuss ideas, editing and sharing photos, for science and email Links to NCCE curriculum: Creating media – Digital photography	
	Copyright and ownership	I can describe why other people's work belongs to them	

Year 3 – Computing				
Term	**Children learn about…**	**Children learn how to…**	**Vocabulary**	
Autumn 2	Using logical reasoning to design, write and debug programs and correct errors in algorithms and programs Using sequence, selection and repetition in programs Programming various forms of input and output Downloaded NCCE unit updated Summer 22	Use pre-made sprite or design a sprite to programme a conversation using Scratch Jr with a simple input and output (e.g. conversation between 2 characters from the Viking/Anglo Saxon struggle – 1 day project) Debug any problems that occur in the process Give well thought-through reasons for errors they find in programs (written by themselves or others/teacher) Explain a simple algorithm in their own words and record algorithms graphically, e.g. as a storyboard Links to NCCE curriculum Yr2: Programming B – An introduction to quizzes – Year 2 – You are not to make a quiz. Year 4 do this. You can write a conversation. It uses the same input / output	Actions blocks change debug, Digital device, evaluate features, input	match modify, outcome, output process program, project, run, sprite, start

Year 3 - Spring 1 Stop-frame animation

- NC objectives: 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.
- Literacy links: draft and write by: in narratives, creating settings, characters and plot
- proof-read for spelling and punctuation errors
- History: The Roman Empire and its impact on Britain

Lesson number	Learning objective	Pupils will:	Vocabulary
1	To explain that animation is a sequence of drawings or photographs	• Discuss whether they think a picture can move. • Learn about simple animation techniques • Create their own animations in the style of flip books (flick books) using sticky notes.	Animation, flip book
2	To relate animated movement with a sequence of images	• Develop this knowledge of animation and apply it to make a stop-frame animation using a tablet.	Stop-frame animation, frame, sequence, image, photograph
3	To plan an animation	• Create a storyboard showing the characters, settings and events that they would like to include in their own stop-frame animation.	Setting, character, events, stop-frame animation, onion skinning
4	To identify the need to work consistently and carefully	• Use tablets to carefully create stop-frame animations, paying attention to consistency.	Stop-frame animation, onion skinning, consistency
5	To review and improve an animation	• Evaluate their animations and try to improve them by creating a brand-new animation based on their feedback.	Evaluation, animation, onion skinning, delete, frame
6	To evaluate the impact of adding other media to an animation	• Add other media and effects into their animations, such as music and text.	Animation, media, import, transition

Year 3 – Spring 2 – Branching databases

- NC objectives: 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

Lesson number	Learning objective	Pupils will:	Vocabulary
1	To create questions with yes/no answers	● Explore questions with yes/no answers and how these can be used to identify and compare objects. ● Create their own yes/no questions, before using these to split a collection of objects into groups.	Attribute, value, questions, table, objects
2	To identify the attributes needed to collect data about an object	● Develop their understanding of using questions with yes/no answers to group objects more than once. ● Learn how to arrange objects into a tree structure and will continue to think about which attributes the questions are related to.	Branching database, database, attribute, value, questions, objects, equal, even, separate
3	To create a branching database	● Develop their understanding of ordering objects/images in a branching database structure. ● Learn how to use an online database tool to arrange objects into a branching database ● Create their own questions with yes/no answers. ● Show that their branching database works through testing.	Branching database, database, attribute, value, questions, objects
4	To explain why it is helpful for a database to be well structured	● Develop their understanding of how to create a well-structured database. ● Use attributes to create questions with yes/no answers and will apply these to given objects. ● Compare the efficiency of different branching databases and will be able to explain why questions need to be in a specific order.	Branching database, attribute, questions, structure, compare, order, organise
5	To plan the structure of a branching database	● Independently plan a branching database by creating a physical representation of one that will identify different types of dinosaur. ● Think about the attributes of objects to write questions with yes/no answers, which will enable them to separate a group of objects effectively. ● Arrange the questions and objects into a tree structure, before testing the structure.	Branching database, attribute, value, question, selecting
6	To independently create an identification tool	● Independently create a branching database to identify different types of dinosaur, based on the paper-based version that they created in Lesson 5. ● Work with a partner to test that their database works, before considering real-world applications for branching databases.	Branching database, attribute, value, questions, information, decision tree

Year 3 Summer 1 – Connecting computers

- NC objectives: 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
- 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
- **Maths (Lesson 1) Number and place value:** solve number problems and practical problems involving these ideas.
- **Art (Lesson 3)** to improve their mastery of art and design techniques, including drawing, painting and sculpture with a range of materials [for example, pencil, charcoal, paint, clay]

Lesson number	Learning objective	Pupils will:	Vocabulary
1	To explain how digital devices function	● Be introduced to the concepts of input, process and output. These concepts are fundamental to all digital devices.	Digital device, input, process, output
2	To identify input and output devices	● Develop their knowledge of the relationship between inputs, processes and outputs ● Apply it to devices and parts of devices that they will be familiar with from their everyday surroundings.	Digital device, input, process, output
3	To recognise how digital devices can change the way that we work	● Apply their learning from Lessons 1 and 2 by using programs in conjunction with inputs and outputs on a digital device. ● Create two pieces of work with the same focus, using digital devices to create one piece of work and non-digital tools to create the other. ● Compare and contrast the two approaches.	Program, digital, non-digital
4	To explain how a computer network can be used to share information	● Explain how and why computers are joined together to form networks.	Connection, network, network switch
5	To explore how digital devices can be connected	● Examine each device's functionality and look at the benefits of networking computers.	Server, wireless access point
6	To recognise the physical components of a network	● Develop their understanding of computer networks. ● Using examples of network infrastructure in a real-world setting, relate them to the activities in Lesson 5.	Network cables, network sockets

Year 3 – Summer 2 Events and actions in programs

- NC objectives: 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

Lesson number	Learning objective	Pupils will:	Vocabulary
1	To explain how a sprite moves in an existing project	● Investigate how characters can be moved using 'events'. ● Analyse and improve an existing project and apply what they have learned to their own projects. ● Extend their learning to control multiple sprites in the same project.	Motion, event, sprite, algorithm, logic
2	To create a program to move a sprite in four directions	● Program a sprite to move in four directions: up, down, left and right. ● Begin by choosing a sprite and sizing it to fit in with a given background. ● Create the code to move the sprite in one direction before duplicating and modifying it to move in all four directions. ● Consider how their project could be extended to prove that their sprite has successfully navigated a maze.	Move, resize, algorithm
3	To adapt a program to a new context	● Use the pen down block to draw lines, building on the movement they created for their sprite in Lesson 2. ● Decide how to set up their project every time it is run.	Extension block, pen up, set up
4	To develop my program by adding features	● Use additional **Pen** blocks. ● Predict the functions of new blocks and experiment with them, before designing features to add to their own projects. ● Add these features to their projects and test their effectiveness.	Pen, design, event, action, algorithm
5	To identify and fix bugs in a program	● Review an existing project against a given design and identify bugs within it. ● Correct the errors, gaining independence as they do so. ● Develop their projects by considering which new setup blocks to use.	Debugging, errors, setup
6	To design and create a maze-based challenge	● Design and create their own projects. ● Use a template to move a sprite around a maze, with the option to leave a pen trail showing where the sprite has moved.	Design, code, setup, test, debug, actions, events

Year 4 – Autumn 2 selection in quizzes

- NC objectives: 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

Lesson number	Learning objective	Pupils will:	Vocabulary
1	To explain how selection is used in computer programs	● Revisit previous learning on 'selection' and identify how 'conditions' are used to control the flow of actions in a program. ● Be introduced to the blocks for using conditions in programs using the Scratch programming environment. ● Modify the conditions in an existing program and identify the impact this has.	Selection, condition, true, false, count-controlled loop
2	To relate that a conditional statement connects a condition to an outcome	● Develop their understanding of selection by using the 'if… then… else…' structure in algorithms and programs. ● Revisit the need to use repetition in selection to ensure that conditions are repeatedly checked. ● Identify the two outcomes in given programs and how the condition informs which outcome will be selected. ● Use this knowledge to write their own programs that use selection with two outcomes.	Selection, condition, true, false, outcomes, conditional statement (the linking together of a condition and outcomes), algorithm, program, debug
3	To explain how selection directs the flow of a program	● Consider how the 'if… then… else…' structure can be used to identify two responses to a binary question (one with a 'yes or no' answer). ● Identify that the answer to the question is the 'condition' and use algorithms with a branching structure to represent the actions that will be carried out if the condition is true or false. ● Learn how questions can be asked in Scratch and how the answer, supplied by the user, is used in the condition to control the outcomes. ● Use an algorithm to design a program that uses selection to direct the flow of the program based on the answer provided. ● Implement their algorithm as a program and test whether both outcomes can be achieved.	Selection, condition, true, false, outcomes, question, answer, algorithm, program, debug
4	To design a program that uses selection	● Use selection to control the outcomes in an interactive quiz. ● Outline the requirements of the task and use an algorithm to show how they will use selection in the quiz to control the outcomes based on the answer given. ● Complete their designs by using design templates to identify the questions that will be asked and the outcomes for both correct and incorrect answers. ● Demonstrate their understanding of how they are using selection to control the flow of the program ● Identify which outcomes will be selected based on given responses.	Task, design, algorithm, input, program, selection, condition, outcomes
5	To create a program that uses selection	● Use the Scratch programming environment to implement the first section of their algorithm as a program. ● Run the first section of their program to test whether they have correctly used selection to control the outcomes and debug their program if required. ● Implementing their algorithm as a program. ● Consider the value of sharing their program with others so that they can receive feedback ● Conclude the lesson by using another learner's quiz and providing feedback on it.	Implement, design, algorithm, program, selection, condition, outcome, test, run
6	To evaluate my program	● Return to their completed programs and identify ways in which the program can be improved. ● Focus on issues where answers similar to those in the condition are given as inputs and identify ways to avoid such problems. ● Consider how the outcomes may change the program for subsequent users and identify how they can make use of 'setup' to provide all users with the same experience. ● Implement their identified improvements by returning to the Scratch programming environment and adding to their programs. ● Conclude the unit by identifying how they met the requirements of the given task and identifying the aspects of the program that worked well, those they improved and areas that could improve further.	Design, algorithm, program, debug, test, setup, selection, condition, outcome

Year 4 – Spring 1 Repetition in shapes

- NC objectives: 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

Lesson number	Learning objective	Pupils will:	Vocabulary
1	To identify that accuracy in programming is important	● Learn the basic Logo commands ● Use their knowledge of them to read and write code.	Program, Turtle, Commands, Code snippet
2	To create a program in a text-based language	● Create algorithms (a precise set of ordered instructions, which can be turned into code) for their initials. ● Implement these algorithms by writing them in Logo commands to draw the letter. ● Debug their code by finding and fixing any errors that they spot.	Algorithm, design, debug, logo
3	To explain what 'repeat' means	● Look at examples of patterns in everyday life. ● Recognise where numbers, shapes and symbols are repeated and how many times repeats occur. ● Create algorithms for drawing a square, using the same annotated diagram as in Lesson 2. ● Use this algorithm to program a square the 'long' way and recognise the repeated pattern within a square. ● Use the repeat command within Logo to program squares the 'short' way.	Pattern, repeat, repetition, count-controlled loop, algorithm, value
4	To modify a count-controlled loop to produce a given outcome	● Work with count-controlled loops in a range of contexts. ● Think about a real-life example, then they will move on to using count-controlled loops in regular 2D shapes. ● Trace code to predict which shapes will be drawn ● Modify existing code by changing values within the code snippet.	Repeat, repetition, count-controlled loop, trace, value
5	To decompose a task into small steps	● Break down everyday tasks into smaller parts and think about how code snippets can be broken down to make them easier to plan and work with. ● Learn to create, name and call procedures in Logo, which are code snippets that can be reused in their programming.	Repeat, Count-controlled loop, Decompose, Procedure
6	To create a program that uses count-controlled loops to produce a given outcome	● Apply the skills that they have learnt in this unit to create a program containing a count-controlled loop. ● Design wrapping paper using more than one shape, which they will create with a program that uses count-controlled loops. ● Create the algorithm, either as an annotated sketch, or as a sketch and algorithm and then implement it as code. ● Debug their work throughout and evaluate their programs against the original brief.	Count-controlled loop, Procedure, Debug, program

Year 4 – Spring 2 photo editing

● NC objectives: 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			
Lesson number	Learning objective	Pupils will:	Vocabulary
1	To explain that the composition of digital images can be changed	● Introduce learners to the concept of editing images. ● Explore when we need to rotate and crop an image as well as how to use an image editor to make these changes. ● Discuss image composition.	Image, edit, digital, crop, rotate, undo, save
2	To explain that colours can be changed in digital images	● Look at the effect that different colours and filters can have on an image. ● Choose appropriate effects to fit a scenario and explain how they made their choices. ● Edit the images using different effects to suit two different scenarios.	Image, adjustments, effects, colours, hue, saturation, sepia, vignette
3	To explain how cloning can be used in photo editing	● Be introduced to the cloning tool and its use in both changing the composition of a photo and photo retouching. ● See how parts of a photo can be removed or duplicated using cloning. ● Consider what parts of an image can be retouched and learn techniques to make this as unnoticeable as possible. ● Consider when it is necessary to edit photographs in this way.	Image, edit, retouch, clone
4	To explain that images can be combined	● Learn how to use different tools to select areas of an image. ● Copy and paste within one image and between two images to produce a combined image. ● Consider when it's appropriate to edit an image and discuss some of the ethics around retouching photos.	Image, edit, select, copy, paste, combine
5	To combine images for a purpose	● Apply all the skills they have learnt in the unit so far. ● Review some images and consider what makes an image look real or made up. ● Plan their own image. ● Choose from a selection of images, open them and edit them to create their own project.	Image, made up, real, composite, cut, copy, paste, alter, background, foreground
6	To evaluate how changes can improve an image	● Review the image that they created in Lesson 5. ● Make changes to their image based on their review. ● Add text to their image to complete it as a publication.	Rotate, crop, zoom, clone, select, copy, paste, undo, font

Year 4 – Summer 1 Data logging

- NC objectives: Use sequence, selection and repetition in programs; work with variables and various forms of input and output
- 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
- **Science – Lower key stage 2/Year 4**: Making systematic and careful observations and, where appropriate, taking accurate measurements using standard units, using a range of equipment, including thermometers and data loggers.
- They should learn how to use new equipment, such as data loggers, appropriately. They should collect data from their own observations and measurements, using notes, simple tables and standard units and help to make decisions about how to record and analyse this data.

Lesson number	Learning objective	Pupils will:	Vocabulary
1	To explain that data gathered over time can be used to answer questions	● Consider what data can be collected and how it is collected. ● Think about data being collected over time. ● Think about questions that can and can't be answered using available data ● Reflect on the importance of collecting the right data to answer questions.	Data, table, layout
2	To use a digital device to collect data automatically	● Build on the idea of collecting data over time and be introduced to the idea of collecting data automatically using computers such as data loggers. ● Be introduced to the concept that computers can capture data from the physical world using input devices called 'sensors'. ● Establish that sensors can be connected to data loggers, which can automatically collect data while not attached to a computer.	Input device, sensor, data logger
3	To explain that a data logger collects 'data points' from sensors over time	● Explore how data loggers work. ● Record data at set moments in time and draw parallels with the data points that a data logger captures at regular intervals. ● Use data loggers away from a computer, then they will connect the loggers to a computer and download the data.	Data logger, logging, data point, interval
4	To recognise how a computer can help us analyse data	● Open an existing data file and use software to find out key information. ● Analyse a data file which is a five-hour log of hot water cooling to room temperature.	Analyse, data set, import, export
5	To identify the data needed to answer questions	● Think about questions that can be answered using collected data. ● Choose a question to focus on and then plan the data logging process that they need to complete. ● Set up the data loggers to check that their plan will work.	Data, data logger, logged, collection
6	To use data from sensors to answer questions	● Access and review the data that they have collected using a data logger. ● Use the data collected to answer the question that they selected in the previous lesson. ● Reflect on the benefits of using a data logger.	Analyse, review, conclusion

Year 4 – Summer 2 Repetition in games

- NC objectives: 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

Lesson number	Learning objective	Pupils will:	Vocabulary
1	To develop the use of count-controlled loops in a different programming environment	● Look at real-life examples of repetition and identify which parts of instructions are repeated. ● Use Scratch, a block-based programming environment, to create shapes using count-controlled loops. ● Consider what the different values in each loop signify. ● Use existing code to modify and create new code. ● Work on reading code and predicting what the output will be once the code is run.	Scratch, programming, sprite, blocks, code, loop, repeat, value
2	To explain that in programming there are infinite loops and count-controlled loops	● Look at different types of loops: infinite loops and count-controlled loops. ● Practise using these within Scratch and think about which might be more suitable for different purposes.	Block, repeat, forever, infinite loop, count-controlled loop, costume
3	To develop a design that includes two or more loops which run at the same time	● Create designs for an animation of the letters in their names. ● Program the animations in Scratch. ● Evaluate their work, considering how effectively they used repetition in their code.	Repetition, forever, infinite loop, count-controlled loop, animate, costume, event block, duplicate
4	To modify an infinite loop in a given program	● Look at an existing game and match parts of the game with the design. ● Make changes to a sprite in the existing game to match the design. ● Look at a completed design and implement the remaining changes in the Scratch game. ● Add a sprite, re-use and modify code blocks within loops and explain the changes made.	Block, repeat, forever, infinite loop, modify, design
5	To design a project that includes repetition	● Look at a model project that uses repetition. ● Design their own games based on the model project, producing designs and algorithms for sprites in the game. ● Share these designs with a partner and have time to make any changes to their design as required.	Infinite loop, count-controlled loop, repetition, design, sprite, algorithm
6	To create a project that includes repetition	● Build their games, using the designs they created in Lesson 5. ● Follow their algorithms, fix mistakes and refine designs in their work as they build. ● Evaluate their work once it is completed and showcase their games at the end.	Repetition, design, algorithm, duplicate, debug, refine, evaluate

Year 5 – Autumn 2 Variables in games

- NC objectives: 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

Lesson number	Learning objective	Pupils will:	Vocabulary
1	To define a 'variable' as something that is changeable	● Be introduced to variables. ● See examples of real-world variables (score and time in a football match) before they explore them in a Scratch project. ● Design and make their own project that includes variables. ● Identify that variables are named and that they can be letters (strings) as well as numbers	Variable, change, name, value
2	To explain why a variable is used in a program	● Understand that variables are used in programs and that they can only hold a single value at a time. ● Complete an unplugged task that demonstrates the process of changing variables. ● Explore why it is important to name variables and apply their learning in a Scratch project in which they make, name and update variables.	Variable, name, value, set, change
3	To choose how to improve a game by using variables	● Apply the concept of variables to enhance an existing game in Scratch. ● Predict the outcome of changing the same change score block in different parts of a program. ● Test their predictions in Scratch. ● Experiment with using different values in variables and with using a variable elsewhere in a program. ● Add comments to their project to explain how they have met the objectives of the lesson.	Variable, set, change, design, event
4	To design a project that builds on a given example	● Design the sprites and backgrounds for their project. ● Design their algorithms to create their program flow.	Design, algorithm, code
5	To use my design to create a project	● Implement the algorithms that they created in Lesson 4. ● Identify variables in an unfamiliar project and learn the importance of naming variables. ● Add another variable to enhance their project.	Task, algorithm, design, artwork, program, project, code, test, debug
6	To evaluate my project	● Build on the project that they created in Lesson 5. ● Consider how they could improve their own projects and make small changes to achieve this. ● Add a variable independently. ● Evaluate each other's projects; they identify features that they liked and features that could be improved.	Improve, evaluate, share

Year 5 – Spring 1 Introduction to vector graphics

- NC objectives: 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.

Lesson number	Learning objective	Pupils will:	Vocabulary
1	To identify that drawing tools can be used to produce different outcomes	● Be introduced to vector drawings and begin to understand that they are made up of simple shapes and lines. ● Use the main drawing tools within the Google Drawings application to create their own vector drawings. ● Discuss how vector drawings differ from paper-based drawings.	Vector, drawing tools, object, toolbar
2	To create a vector drawing by combining shapes	● Identify the shapes that are used to make vector drawings. ● Explain that each element of a vector drawing is called an object. ● Create their own vector drawing by moving, resizing, rotating and changing the colours of a selection of objects. ● Duplicate the objects to save time.	Vector drawing, object, move, resize, colour, rotate, duplicate/copy
3	To use tools to achieve a desired effect	● Increase the complexity of their vector drawings and use the zoom tool to add detail to their work. ● Be shown how grids and resize handles can improve the consistency of their drawings. ● Use tools to modify objects to create a new image.	Zoom, select, rotate, object, align, resize, modify
4	To recognise that vector drawings consist of layers	● Gain an understanding of layers and how they are used in vector drawings. ● Discover that each object is built on a new layer and that these layers can be moved forwards and backwards to create effective vector drawings.	Layers, object, order
5	To group objects to make them easier to work with	● Select and duplicate multiple objects at a single time. ● Develop this skill further by learning how to group multiple objects to make them easier to work with. ● Use this knowledge to group and ungroup objects, in order to make changes to and develop their vector drawings.	Copy, paste, group, ungroup, duplicate, object, vector drawing, reuse
6	To apply what I have learned about vector drawings	● Use the skills they have gained in this unit to create a vector drawing for a specific purpose. ● Reflect on the skills they have used to create the vector drawing and think about why they used the skills they did. ● Begin to compare vector drawings to freehand paint program drawings.	Reflection, vector drawing

Year 5 – Spring 2 Systems and searching

- NC objectives: 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

Lesson number	Learning objective	Pupils will:	Vocabulary
1	To explain that computers can be connected together to form systems	● Be introduced to the concept of a system. ● Begin to understand that components can work together to perform a task. ● Explore how digital systems can work and learn about physical and electronic connections.	System, connection, digital, input, process, output
2	To recognise the role of computer systems in our lives	● Consider how larger computer systems work. ● See how devices and processes are connected and reflect on how computer systems can help them.	System, connection, digital, input, process, output
3	To identify how to use a search engine	● Be introduced to a range of search engines. ● Be given the opportunity to explain how to search, before they write and test instructions. ● Learn that searches do not always return the results that someone is looking for and refine their searches accordingly. ● Be introduced to the two most common methods of searching: using a search engine and using the address bar.	Search, search engine, refine
4	To describe how search engines select results	● Gain an understanding of why search engines are necessary to help them find things on the World Wide Web. ● Conduct their own searches and break down, in detail, the steps needed to find things on the web. ● Emulate web crawlers to create an index of their own classroom. ● Consider why some searches return more results than others.	Index, crawler, bot, search engine
5	To explain how search results are ranked	● Create paper-based webpages on a topic that they are familiar with. ● Discover how their webpages would rank when searching for keywords relating to their content.	Ordering, ranking, search engine, links, algorithm, search engine optimisation (SEO)
6	To recognise why the order of results is important and to whom	● Explore how someone performing a web search can influence the results that are returned ● Understand how content creators can optimise their sites for searching. ● Explore some of the limitations of searching and discuss what cannot be searched.	Searching, search engine, web crawler, content creator, selection, ranking

Year 5 – Summer 1 / Summer 2 selection in physical computing

- NC objectives: 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
- Science: Construct a simple series electrical circuit, identifying and naming its basic parts, including cells, wires, bulbs, switches and buzzers
- Design and Technology: Design: Generate, develop, model and communicate their ideas through discussion, annotated sketches, cross-sectional and exploded diagrams, prototypes, pattern pieces and computer-aided design
- Make: Select from and use a wider range of tools and equipment to perform practical tasks [for example, cutting, shaping, joining and finishing], accurately
- Select from and use a wider range of materials and components, including construction materials, textiles and ingredients, according to their functional properties and aesthetic qualities
- Evaluate: Evaluate their ideas and products against their own design criteria and consider the views of others to improve their work
- Technical knowledge: Understand and use electrical systems in their products [for example, series circuits incorporating switches, bulbs, buzzers and motors]
- Apply their understanding of computing to program, monitor and control their products

Lesson number	Learning objective	Pupils will:	Vocabulary
1	To control a simple circuit connected to a computer	• Become familiar with the Crumble controller and the programming environment used to control it. • Connect a Sparkle to a Crumble and then program the Crumble to make the Sparkle flash different colour patterns. • Use infinite loops, which were introduced to the learners in the previous school year.	Microcontroller, components, connection, infinite loop
2	To write a program that includes count-controlled loops	• Connect a Sparkle and a motor to the Crumble controller. • Design sequences of actions for these components. • Apply their understanding of repetition by using count-controlled loops when implementing their design as a program.	Microcontroller, output component, motor, repetition, count-controlled loop
3	To explain that a loop can stop when a condition is met	• Be introduced to conditions and how they can be used in programs to control their flow. • Identify conditions in statements, stating if they are true or false. • Be introduced to a Crumble switch and learn how it can provide the Crumble controller with an input that can be used as a condition. • Explore how to write programs that use an input as a condition.	Microcontroller, Crumble controller, components, switch, motor, LED, Sparkle, crocodile clips, connect, battery box, program, condition
4	To explain that a loop can be used to repeatedly check whether a condition has been met	• Develop their understanding of how the flow of actions in algorithms and programs can be controlled by conditions. • Be introduced to selection and then represent conditions and actions using the 'if…then…' structure. • Create algorithms that include selection. • Use their algorithms to guide their program writing. • See that infinite repetition is required to repeatedly check if a condition has been met.	Input, output, selection, condition, action
5	To design a physical project that includes selection	• Apply their understanding of microcontrollers and selection when designing a project to meet the requirements of a given task. • Identify how selection might be used in real-world situations. • Consider how they can apply this knowledge to design their project. • Produce design sketches to show how their model will be made and how they will connect the microcontroller to its components.	Selection, condition, action, repetition
6	To create a program that controls a physical computing project	• Identify how they are going to use selection before writing an algorithm to meet the requirements of the given task. • Implement their algorithms as code. • Run their programs to identify any bugs and then return to the code or algorithm to debug it where necessary. • Evaluate their designs.	Selection, condition, action, repetition, debug

Year 5 – Summer 1 / Summer 2 Flat-file databases Summer 1 2024 only

- NC objectives: 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

Lesson number	Learning objective	Pupils will:	Vocabulary
1	To use a form to record information	● Create a paper version of a record card database. ● Create a data set, with each learner creating eight to ten cards linked to a theme, e.g. animals. ● Complete records for each of the animals in their database and then they will physically sort the cards to answer questions about the data.	Database, data, information, record, field, sort, order, group
2	To compare paper and computer-based databases	● Examine how data can be recorded and viewed. ● Learn that a database consists of 'records' and that each record contains 'fields'. ● Order records in different ways and compare this database to the paper database they created in Lesson 1.	Database, data, field, record, sort, order
3	To outline how you can answer questions by grouping and then sorting data	● Investigate how records can be grouped, using both the paper record cards created in Lesson 1 and a computer-based database from J2E. ● Use 'grouping' and 'sorting' to answer questions about the data.	Database, record, field, group, search, sort, order
4	To explain that tools can be used to select specific data	● Develop their search techniques to answer questions about the data. ● Use advanced techniques to search for more than one field and will practise doing this through both unplugged methods (without using computers) and using a computer database.	Database, record, field, value, search, criteria
5	To explain that computer programs can be used to compare data visually	● Consider what makes a useful chart and how charts can be used to compare data. ● Create charts from their data in order to answer questions about it.	Database, record, field, graph, chart, axis, compare, filter
6	To use a real-world database to answer questions	● Use a real-life database to ask questions and find answers in the context of a flight search based on set parameters. ● Take on the role of a travel agent and present their findings, showing how they arrived at their chosen options. ● Presentations may be given between groups of learners, or by each group to the whole class, depending on the time available	Database, field, record, graph, chart, presentation

Year 6 – Computing			
Term	Children learn about…	Children learn how to…	Vocabulary
Autumn 2	Using search technology effectively How search results are selected and ranked and how to be discerning in evaluating digital content Computer networks Downloaded NCCE Unit Updated Summer 22	Develop their understanding of computer systems and how information is transferred between systems and devices. Consider small-scale systems as well as large-scale systems. Explain the input, output, and process aspects of a variety of different real-world systems. Find information on the World Wide Web, through learning how search engines work (including how they select and rank results) and what influences searching, and through comparing different search engines. Investigate different methods of communication, before focusing on internet-based communication. Evaluate which methods of internet communication to use for particular purposes. Links to NCCE curriculum: Computing systems and networks – Sharing information Links to NCCE curriculum Yr 6: Computing systems and networks – Communication	accelerometer bot, breadcrumb trail, browser, collaboration compass, copyright, crawler, Domain Name Server (DNS) embed external link, fair use Google Sites header, home page, Hyperlink, Hypertext Markup Language (HTML) if then else implication, Index, Internet Protocol (IP) address, layout, logo, media, Micro:bit navigation, protocol, ranking, refine remix, search engine optimisation (SEO) sensing, step counter subpage
Spring	Selecting, using and combining a variety of software Designing and creating a range of programs, systems and content that accomplish given goals Collecting, analysing, evaluating and presenting data and information Using search technology effectively and how to be discerning in evaluating digital content Downloaded NCCE Unit Updated Summer 22	Create a presentation on a range of different media, combining multiple devices with greater difference Use Googlesite to create a presentation on a given topic, making sure they use different software tools Use a range of sites to find information on a particular topic Judge whether the information on different sites is accurate and use a variety of sources to support their information Explore copyright of information and what things they should and shouldn't use in their work Describe the implications of plagairising Links to NCCE curriculum Yr 6:Creating media – Web page creation	
	Copyright and ownership Managing information online	I can demonstrate how to make references to and acknowledge sources I have used from the internet I can demonstrate how to analyse and evaluate the validity of 'facts' and information and I can explain why using these strategies are important.	
Summer 2	Using logical reasoning to design, write and debug programs Using sequence, selection and repetition in programs Programming various forms of input and output Downloaded NCCE Unit Updated Summer 22	Bring together elements of all the four programming constructs: sequence, repetition, selection, and variables. Children should code using a new form of coding such as HTML or Python. Discovery coding can be used here - See Syed for login information Give clear and precise explanations of how a number of algorithms work When given an algorithm for a particular purpose, identify and correct possible errors and explain why they believe their correction will work Use all of the constructs of programming in a different, but still familiar environment, while also utilising a physical device — the micro:bit. (e.g. MakeyMakey?) Links to NCCE curriculum Yr 6: Programming B – Sensing	