Computing Curriculum

Computing Intent and Knowledge Progression Map

A Member of Exceed Learning Partnership
• EVERY CHILD • EVERY CHANCE • EVERY DAY •

Sheep Dip Lane Academy

Introduction

This information outlines the knowledge, language and concepts that should be taught in Computing. It includes:

- A summary of the Computing knowledge and principles that underpin our approach
- Long Term Sequence (curriculum map) for Computing

It is influenced by documents and research, including https://www.gov.uk/government/publications/research-review-series-computing/research-review-series-computing.

Intent

It is our intention that through studying Computing, pupils become more expert as they progress through the curriculum, accumulating, connecting and making sense of the substantive knowledge and disciplinary knowledge/skills.

1. **declarative knowledge (know that)** - this is the subject knowledge and explicit vocabulary used to learn about the content. Common misconceptions are explicitly revealed as non-examples and positioned against known and accurate content.

2. **procedural knowledge (know how)** – this is knowing how to collect, use, interpret, understand and evaluate learning through the Computing knowledge that is taught. It is not assumed that pupils will acquire these skills by luck or hope.

We acknowledge that the Computing curriculum is split into three main pillars and content areas:

Computer science **Information technology** **Digital literacy**

'However, these pillars do not sit separately from each other. Knowledge from each pillar complements the others and some subject content only exists at the interplay between these 3 pillars.' Therefore all learning outcomes are planned out under four strands, which provides categories and an organised view of content to encapsulate the discipline of computing:

- Computer Systems and Networks- Understand what a computer is, and how its constituent parts function together as a whole, and understand how networks can be used to retrieve and share information, and how they come with associated risks

- Programming- Create software to allow computers to solve problems

- Creating Media- Select and create a range of media including text, images, sounds, and video

- Data and Information- Understand how data is stored, organised, and used to represent real-world artefacts and scenarios

Embedded within these main strands, are the following Digital Literacy skills that are crucial for pupils use of technology in the world:

- **Effective use of tools** — Use software tools to support computing work
- **Impact of technology** — Understand how individuals, systems, and society as a whole interact with computer systems
- **Safety and security** — Understand risks when using technology, and how to protect individuals and systems

Computing is planned so that the retention of knowledge is much more than just 'in the moment knowledge', which can be seen in the way in which we structure our lessons.

The cumulative nature of the curriculum is made memorable by the implementation of Bjork's desirable difficulties, including retrieval and spaced retrieval practice, word building and deliberate practice tasks. This powerful interrelationship between structure and research-led practice is designed to increase substantive knowledge and accelerate learning within and between study modules. That means the foundational knowledge of the curriculum is positioned to ease the load on the working memory: new content is connected to prior learning. The effect of this cumulative model supports opportunities for children to associate and connect significant computing concepts, over time, and with increasing expertise and knowledge.

Our Computing curriculum has sequenced the national curriculum into meaningful and connected 'chunks' of content to reduce the load on the working memory as well as creating coherent and strong long-term memories. The sequence of substantive and disciplinary knowledge enables pupils to become 'more expert' with each study and grow an ever broadening and coherent mental model of the subject. This guards against superficial, disconnected and fragmented Computing knowledge and weak disciplinary knowledge. High frequency, multiple meaning words (Tier 2) are taught explicitly and help make sense of subject specific words (Tier 3).

Progression Overview

Early Years

As young children take part in a variety of tasks with digital devices, such as moving a Bee Bot around a classroom, they will already be familiar with the device before being asked to undertake tasks related to the key stage one (KS1 - ages 5 - 7 years) computing curriculum, such as writing and testing a simple program. Not only will children be keen to again use a device they had previously enjoyed using, their cognitive load will also be reduced, meaning they are more likely to succeed when undertaking activities linked to the next stage in their learning. Within the revised EYFS statutory framework, there are opportunities within each area of the framework to enable practitioners to effectively prepare children for studying the computing curriculum.

Implementation

We implement our intent using NCCE Teach Computing programme of study. The Teach Computing Curriculum (ncce.io/tcc) is a comprehensive collection of materials produced to support teaching, facilitating the delivery of the entire English computing curriculum. All content is free, and editable under the Open Government Licence (OGL — ncce.io/ogl), ensuring that the resources can be tailored to each individual teacher and school setting. The materials are suitable for all pupils irrespective of their skills, background, and additional needs. 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. The Teach Computing Curriculum has been designed to reduce teacher workload. To ensure this, the Teach Computing Curriculum includes all the resources a teacher needs, covering every aspect from planning, to progression mapping, to supporting materials.

Inclusive and ambitious

The Teach Computing Curriculum has 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.

Research-informed

The subject of computing is much younger than many other subjects, and as such, there is still a lot more to learn about how to teach it effectively. To ensure that teachers are as prepared as possible, the Teach Computing Curriculum builds on a set of pedagogical principles (see the 'Pedagogy' section of this document), which are underpinned by the latest computing research, to demonstrate effective pedagogical strategies throughout. To remain up-to-date as research continues to develop, every aspect of the Teach Computing Curriculum is reviewed each year and changes are made as necessary.

Pedagogy

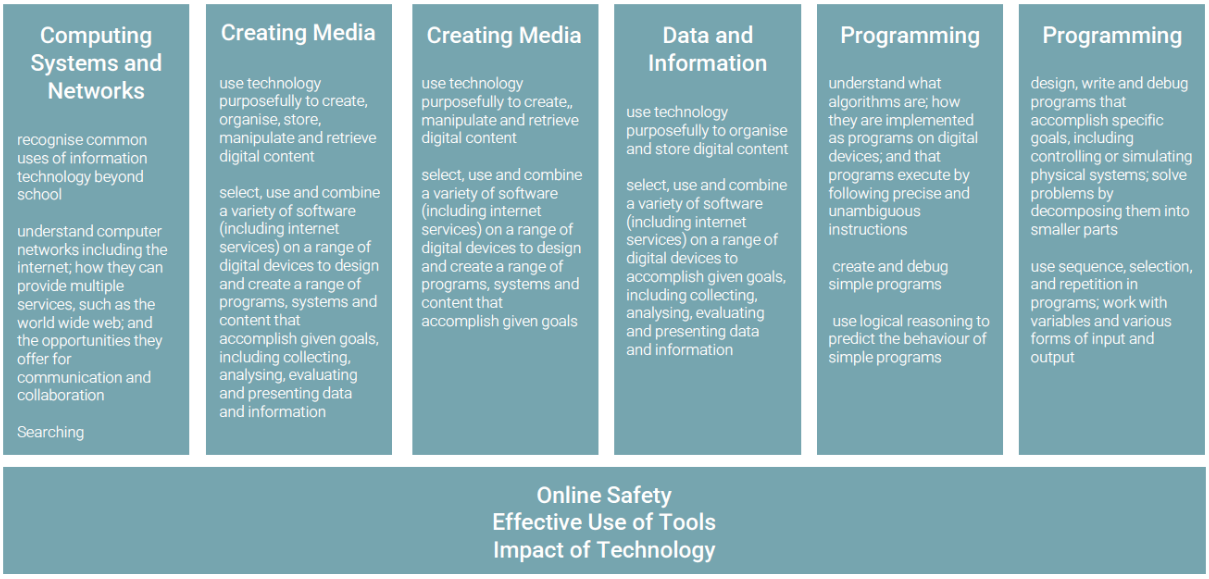
The National Centre for Computing Education's pedagogical approach consists of 12 key principles underpinned by research: each principle has been shown to contribute to effective teaching and learning in computing. It is recommended that computing teachers use their professional judgement to review, select, and apply relevant strategies for their pupils. These 12 principles are embodied by the Teach Computing Curriculum, and examples of their application can be found throughout the units of work at every key stage. Beyond delivering these units, you can learn more about these principles and related strategies in the National Centre for Computing Education pedagogy toolkit (ncce.io/pedagogy).

- **Lead with concepts** Support pupils in the acquisition of knowledge, through the use of key concepts, terms, and vocabulary, providing opportunities to build a shared and consistent understanding. Glossaries, concept maps, and displays, along with regular recall and revision, can support this approach.
- **Work together** Encourage collaboration, specifically using pair programming and peer instruction, and also structured group tasks. Working together stimulates classroom dialogue, articulation of concepts, and development of shared understanding.
- **Get hands-on** Use physical computing and making activities that offer tactile and sensory experiences to enhance learning. Combining electronics and programming with arts and crafts (especially through exploratory projects) provides pupils with a creative, engaging context to explore and apply computing concepts.
- **Model everything** Model processes or practices — everything from debugging code to binary number conversions — using techniques such as worked examples and live coding. Modelling is particularly beneficial to novices, providing scaffolding that can be gradually taken away.
- **Foster program comprehension** Use a variety of activities to consolidate knowledge and understanding of the function and structure of programs, including debugging, tracing, and Parson's Problems. Regular comprehension activities will help secure understanding and build connections with new knowledge.
- **Create projects** Use project-based learning activities to provide pupils with the opportunity to apply and consolidate their knowledge and understanding. Design is an important, often overlooked aspect of computing. Pupils can consider how to develop an artefact for a particular user or function, and evaluate it against a set of criteria.
- **Add variety** Provide activities with different levels of direction, scaffolding, and support that promote learning, ranging from highly structured to more exploratory tasks. Adapting your instruction to suit different objectives will help keep all pupils engaged and encourage greater independence.
- **Challenge misconceptions** Use formative questioning to uncover misconceptions and adapt teaching to address them as they occur. Awareness of common misconceptions alongside discussion, concept mapping, peer instruction, or simple quizzes can help identify areas of confusion.
- **Make concrete** Bring abstract concepts to life with real-world, contextual examples, and a focus on interdependencies with other curriculum subjects. This can be achieved through the use of unplugged activities, proposing analogies, storytelling around concepts, and finding examples of the concepts in pupils' lives.

- **Structure lessons** Use supportive frameworks when planning lessons, such as PRIMM (Predict, Run, Investigate, Modify, Make) and (Use-Modify-Create). These frameworks are based on research and ensure that differentiation can be built in at various stages of the lesson.
- **Read and explore code first** When teaching programming, focus first on code ‘reading’ activities, before code writing. With both block-based and text-based programming, encourage pupils to review and interpret blocks of code. Research has shown that being able to read, trace, and explain code augments pupils’ ability to write code.

Curriculum Sequences

Our computing knowledge and skills are covered through the NCCE programme to ensure that children are taught the appropriate content to support their development. Our teaching of Computing is progressive. The Teach Computing curriculum is a spiral curriculum with units building progressively from year to year. The curriculum is structured into 6 units for each year group, and each unit is broken down into lessons. Below shows the 6 units, which are covered:



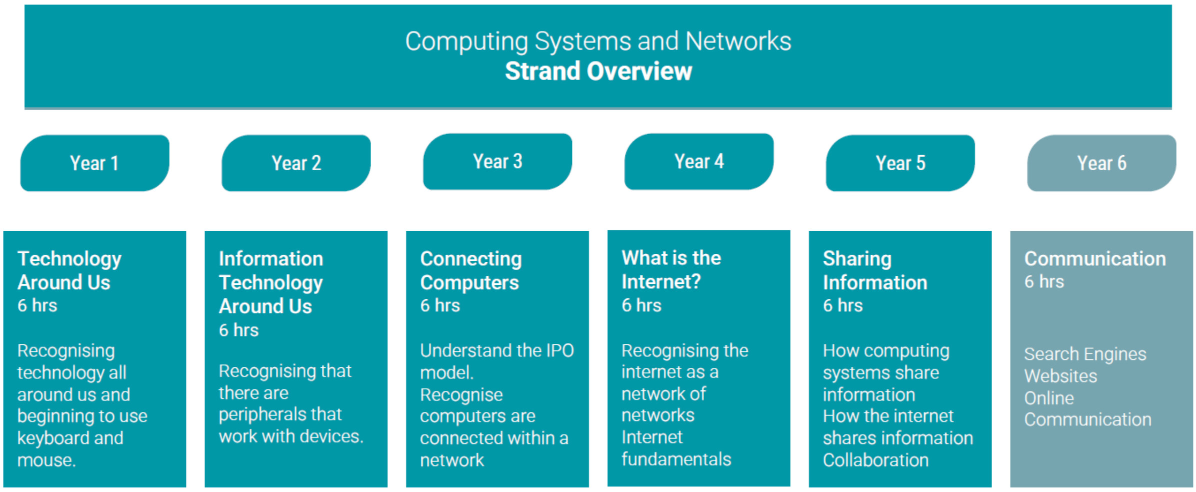
Computing Systems and Networks	Creating Media	Creating Media	Data and Information	Programming	Programming
recognise common uses of information technology beyond school understand computer networks including the internet; how they can provide multiple services, such as the world wide web; and the opportunities they offer for communication and collaboration Searching	use technology purposefully to create, organise, store, manipulate and retrieve 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 purposefully to create,, manipulate and retrieve 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	use technology purposefully to organise and store digital content select, use and combine a variety of software (including internet services) on a range of digital devices to accomplish given goals, including collecting, analysing, evaluating and presenting data and information	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	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

Online Safety
Effective Use of Tools
Impact of Technology

Computer Science

Computer Systems and Networks:

The Computer Systems and Network unit build on the prior learning from the year before and is sequenced progressively throughout the primary phase. The graphics below show how the unit spirally progress from year to year.

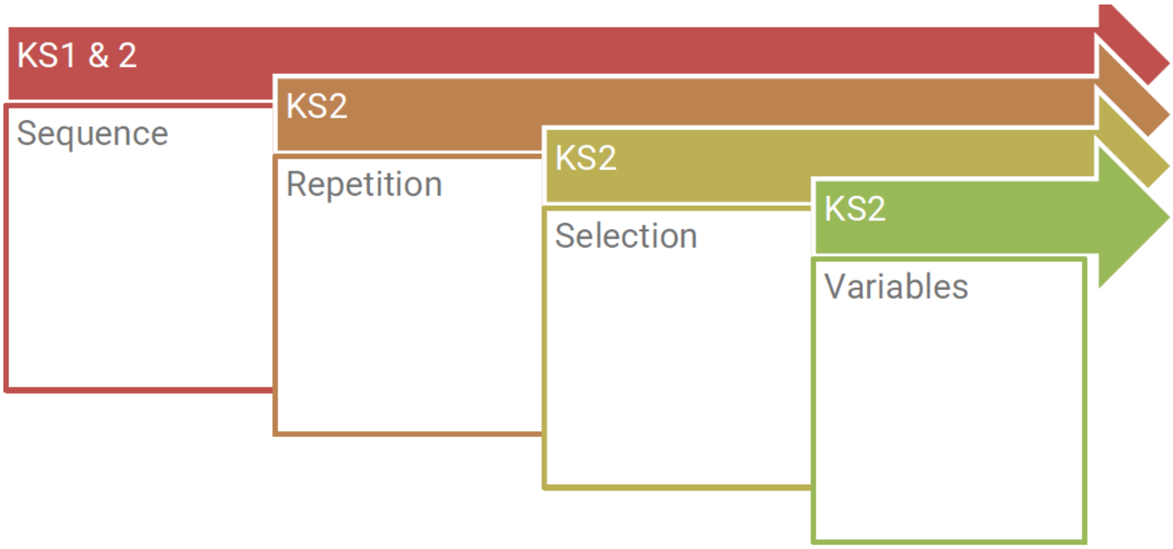


Computing Systems and Networks
Strand Overview

Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Technology Around Us 6 hrs Recognising technology all around us and beginning to use keyboard and mouse.	**Information Technology Around Us** 6 hrs Recognising that there are peripherals that work with devices.	**Connecting Computers** 6 hrs Understand the IPO model. Recognise computers are connected within a network	**What is the Internet?** 6 hrs Recognising the internet as a network of networks Internet fundamentals	**Sharing Information** 6 hrs How computing systems share information How the internet shares information Collaboration	**Communication** 6 hrs Search Engines Websites Online Communication

Programming: Our curriculum incorporates two programming units: Programming A and Programming B, to highlight the emphasis of Computer Science statements in the national curriculum. Programming is revisited twice within a year, with unit B progressing from what they have learnt from unit A. Within Key Stage 1, they look at two different formats:

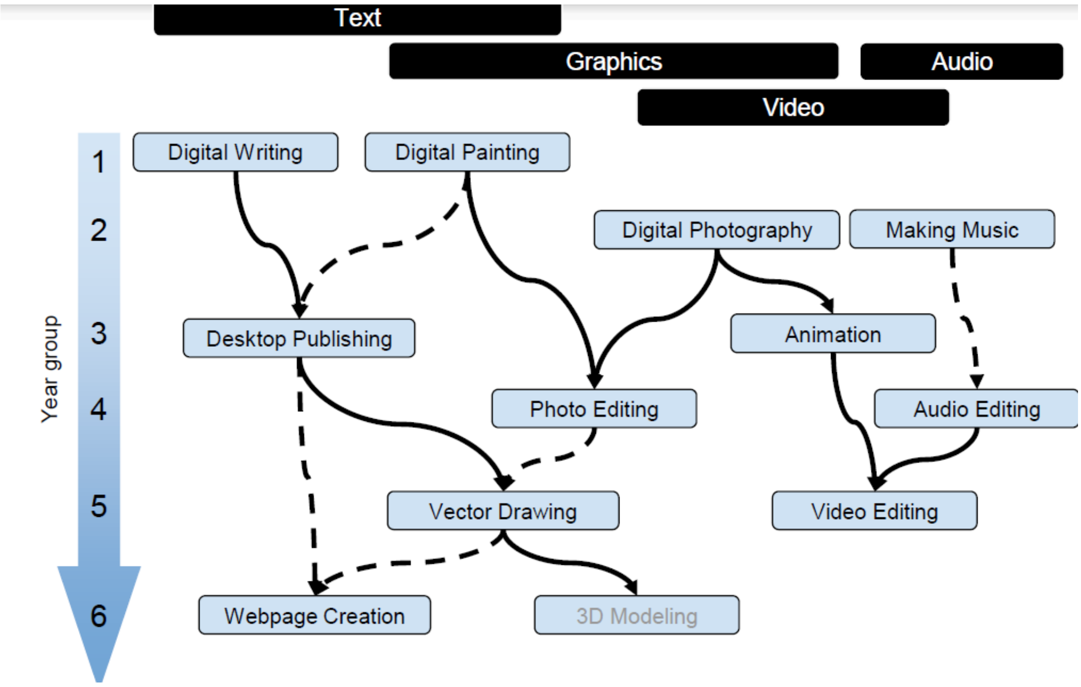
Beebots and Scratch Junior. The learning from Year 1 is built upon in Year 2. In Key Stage 2, the national curriculum statement *"use sequence, selection and repetition in programmes; work with variables and various forms of input and output"* has been identified as a key area, where progression of skills is needed and therefore, the curriculum organisation is sequenced to ensure this. For example, Year 3 focus on sequence, Year 4 focus on repetition, Year 5 on selection and Year 6 on variables. Each year group looks at their skills in depth and masters these over the two units.



Information Technology

Creating Media

Our Creating Media units focus on four main themes: text, graphics, video and audio. Rather than revisiting the same software each year, the curriculum is sequenced to revisit skills learnt prior and apply these using new software, with each year group delivering two units.



Data and Information

The Data and Information unit works in the same way as the Computer Systems and Networks strand, with a unit per year group, looking at different forms of presenting data through Key Stage 2.

Online safety

The unit overviews for each unit show the links between the content of the lessons and the national curriculum and Education for a Connected World framework (ncce.io/efacw). These references have been provided to show where aspects relating to online safety, or digital citizenship, are covered within the Teach Computing Curriculum.

Lesson Structure

Lessons typically are split into six phases:

- CONNECT This provides an opportunity to connect the lesson to prior learning from a previous module or lesson. Teachers return children's attention to the previous lesson's knowledge note/the big idea for the learning module, including key vocabulary. Examples of thinking harder routines include Flick Back 5, Recap questions, Quizzing. Retrieval practice allows all pupils to take time to remember things and activate their memories. Quizzing allows questions to be asked and allows pupils to carry out retrieval practice. Cumulative quizzing, allows for a few questions to be asked each lesson, which are built upon the previous lesson.
- **EXPLAIN** This is the explicit teaching that needs to take place. Teachers should ensure they are clear what they want children to know and remember. They plan for and explicitly address common misconceptions so they can address these in lessons as they arise. They should be clear about the declarative and procedural knowledge and the vocabulary that they want children to understand in the session. This can be developed using key information, facts, and images so that explanations are precise.
- EXAMPLE Providing pupils with high-quality examples is essential for learning. Pupils need to see worked examples. My turn, our turn, your turn is a technique that can be used to explicitly teach vocabulary and new concepts. Prepared examples should be carefully planned and need to be evident in teaching. An example in computing could be demonstrating how to create an algorithm (set of instructions that tells the device what to do), before creating an algorithm for Bee Bot or in Scratch together.
- ATTEMPT Guiding pupil practice allows pupils to rehearse, rephrase and elaborate their learning. Children need the chance to attempt and verbalise their understanding. Children's own attempts are what help them to secure their understanding. Children need to have time to struggle and understand for themselves. This is not necessarily something that is recorded in floor books. This phase provides opportunities for teachers to check in with pupils to see who may need more challenge/support/scaffolds and if any misconceptions have arisen that need to be addressed. Extending the previous computing example, pupils could practise creating their own algorithms.
- APPLY This is where pupils would typically begin to record in individual pupil activity sheets, on Seesaw or in floor books in KS1 or on digital floor books in KS2. The number of scaffolds may vary.
- CHALLENGE Teachers get the children to interrogate their learning - summarise, explain, compare and contrast. Tools are built into routines to reduce overload and allow for hard thinking. These can be adapted for children based on their individual needs.

Impact

In order to identify the impact our curriculum is having on our pupils, we check the extent to which learning has become **permanently embedded** in children's long-term memory in addition to looking for **excellence** in their outcomes. We use four main tools to quality assure the implementation and impact of our curriculum:

- **Learning observations/through drop ins** help to evaluate subject knowledge, explanations, expectations, opportunities to learn, pupil responses, participation and relationships.
- **Professional growth models** help to improve staff subject knowledge and evidence informed practice such as retrieval and spaced practice, interleaving and explicit instruction techniques.
- **Assessment and achievement** articulate the outcomes from tasks and tests, how well the content is understood and what the strengths and limitations are; it informs what to do next.
- **Pupil Book Studies** help to evaluate curriculum structures, teaching methods, pupil participation and response through a dialogic model.

When undertaking these we ask the following key questions:

- How well do pupils remember the content that they have been taught?
- Do books and pupil discussions radiate excellence?
- Does learning 'travel' with pupils and can they deliberately reuse it in more sophisticated contexts?

Teachers employ a range of strategies both at and after the point of teaching to check the impact of their teaching on the permanence of pupils' learning. These include: retrieval practice, vocabulary use and application, deliberate practice and rephrasing of taught content, cumulative quizzing within the learning sequence, summarising and explaining the learning question from the sequence, tests and quizzes. Teachers use information from tasks, tests, pupil book studies and other monitoring to support learning by responding to the gap between where pupils are and where they need to be. In lessons, they adapt explanations and examples to address misconceptions and provide additional practice or challenge where required. After lessons or tests, they analyse pupils' responses to identify shared and individual gaps in learning and misconceptions. Teachers then adjust subsequent planned teaching in response.

We use **summative assessment** is 'to provide an accurate shared meaning without becoming the model for every classroom activity' (Christodolou, 2017). If our curriculum is effective, it will lead to improvements in summative assessments over time. Teacher assessment judgements are against an agreed assessment model (the curriculum). A pupil working at age-related expectations should be able to meet the success criteria for each lesson by the end of the unit. We make summative judgements annually and report these to the subject leader.

Pupil book study is used as a method to quality assure our curriculum by talking to the children and looking in pupils' books. We do this after content has been taught to see the extent to which pupils are knowing more, remembering more and able to do more. In preparation, we review the planned content, knowledge and vocabulary, so that conversations with pupils are meaningful and focused on what has been taught. When looking at books, we look at the content and knowledge, teaching sequence and vocabulary. We also consider pupils' participation and consider the explanations and models used, the tasks the pupils are asked to do, the ability to answer carefully selected questions and retrieve information and the impact of written feedback. We ask careful questions that probe their knowledge, understanding and skills.

The Subject Leader undertakes a range of activities to understand what the curriculum looks like across the school and how well pupils know more, remember more and can do more as a result. In addition to the above tools, they use learning walks, planning reviews and book looks. They use their findings to support teachers to improve how they implement subjects and to make recommendations about the suitability of the intent for their subject. The Subject Leader formally reports on impact of the curriculum termly to the Curriculum Leader, Principal and Governors.

Computing Long Term Overview

	Autumn Term		Spring Term		Summer Term	
	Computer Systems and Networks	Data and information	Programming A	Creating Media	Programming B	Creating Media
EYFS	**Technology Around Us** Recognising what technology is and what technology we have at home and in school	**Grouping Data** Recognising that things can be labelled, grouped and compared.	**Programming** Knowing that things can be labelled, grouped and compared.	**Creating media** Making simple marks on a device, using a painting tool. 2Paint a picture and/or http://paintz.app Recording sounds and speech using a microphone and device.	**Programming** Ordering and sequencing a range of things including stories.	**Creating media** Knowing and using some letters on a keyboard. Knowing how to take photographs using a device.
KS1 Year 1	**Technology Around Us** Recognising technology in school and using it responsibly http://paintz.app	**Grouping Data** Exploring object labels, then using them to sort and group objects by properties Google Docs	**Moving a Robot** Writing short algorithms and programs for floor robots and predicting program outcomes Bee-Bots	**Digital Painting** Choosing appropriate tools in a program to create art and making comparisons with working non-digitally http://paintz.app	**Programming Animations** Designing and programming the movement of a character on screen to tell stories Scratch junior app iPad and Chrome extension https://jfo8000.github.io/ScratchJr-Desktop/	**Digital Writing** Using a computer to create and format text before comparing to writing non-digitally Google docs

KS1 Year 2	**Information Technology Around Us** Identifying IT and how its responsible use improves our world in school and beyond	**Pictograms** Collecting data in tally charts and using attributes to organise and present data on a computer JIT5 (j2e.com)	**Robot Algorithms** Creating and debugging programs and using logical reasoning to make predictions Bee-Bots	**Digital Photography** Capturing and changing digital photographs for different purposes iPad photo app using edit function	**Programming Quizzes** Designing algorithms and programs that use events to trigger sequences of code to make an interactive quiz Scratch junior app iPad and Chrome extension https://jfo8000.github.io/ScratchJr-Desktop/	**Making Music** Using a computer as a tool to explore rhythms and melodies before creating a musical composition Chrome Music Lab Chrome Music Lab (chromeexperiments.com)
LKS2 Year 3	**Connecting Computers** Identifying that digital devices have inputs, processes and outputs and how devices can be connected to make networks http://paintz.app	**Branching Database** Building and using branching databases to group objects using yes/no questions JIT5 (j2e.com)	**Sequencing Sounds** Creating sequences in a block-based programming language to make music Scratch - Imagine, Program, Share (mit.edu)	**Stop-frame Animation** Capturing and editing digital still images to produce a stop-frame animation that tells a story iPad Stop motion studio	**Events and Actions in Programs** Writing algorithms and programs that use a range of events to trigger sequences of actions Scratch - Imagine, Program, Share (mit.edu)	**Desktop Publishing** Creating documents by modifying text, images and page layouts for a specific purpose Microsoft Publisher
LKS2 Year 4	**The Internet** Recognising the internet as a network of networks including the WWW and why we should evaluate online content	**Data Logging** Recognising how and why data is collected over time, before using data loggers to carry out an investigation Data loggers or	**Repetition in Shapes** Using a text- based programming language to explore count-controlled loops when drawing shapes FMS Logo or Turtle Academy Playground	**Audio Production** Capturing and editing audio to produce a podcast, ensuring that copyright is considered Audacity	**Repetition in Games** Using a clock-based programming language to explore count-controlled and infinite loops when creating a game Scratch - Imagine, Program, Share (mit.edu)	**Photo Editing** Manipulating digital images and reflecting on the impact of changes and whether the required purpose is fulfilled paint.net

		Arduino Science Journal iPad app				
UKS2 Year 5	**Systems and Searching** Recognising IT systems around us and how they allow us to search the internet	**Flat-file Database** Using a database to order data and create charts to answer questions https://www.j2e.com/database/	**Selection in Physical Computing** Exploring conditions and selection using a programmable microcontroller Crumbles	**Video Production** Planning, capturing and editing video to produce a short film Google drawing Chrome books Or Microsoft Word laptops	**Selection in Quizzes** Exploring selection in programming to design and code an interactive quiz Scratch online Scratch - Imagine, Program, Share (mit.edu)	**Vector Drawings** Creating images in a drawing program by using layers and groups of objects iPad Video editing iMovie
UKS2 Year 6	**Communication and Collaboration** Identifying and exploring how data is transferred and information is shared online Microsoft PowerPoint	**Introduction to Spreadsheets** Answering questions by using spreadsheets to organise and calculate data Microsoft Excel laptops or Google Sheets chrome books	**Variables in Games** Exploring variables when designing and coding a game Scratch - Imagine, Program, Share (mit.edu)	**Webpage Creation** Designing and creating webpages, giving consideration to copyright , aesthetics and navigation Tinker Cad	**Sensing Movement** Designing and coding a project that captures inputs from a physical device Micro bits	**3D Modelling** Planning, developing and evaluating 3D computer models physical **Chromebooks/laptops and pupils need a google email to login google sites.**

Computing Knowledge Progression Map 2023-2024

	EYFS	Y1	Y2	Y3	Y4	Y5	Y6
NC objectives		• Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions • Create and debug simple programs Use logical reasoning to predict the behaviour of simple programs • Use technology purposefully to create, organise, store, manipulate and retrieve digital content • Recognise common uses of information technology beyond school • Use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies.		• 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 • 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			
Autumn 1 Computer Systems and Networks	**Technology Around Us** (Computer systems and algorithms) -Pupils know the different types of **technology.** -Pupils know the main parts of a **computer (keyboard, screen, mouse, trackpad).** Pupils know how to control a **cursor** using a **mouse/trackpad.**	**Technology Around Us** (Computer systems and algorithms) -Pupils know and can identify different types of **technology.** -Pupils know the main parts of a computer (on/off **switch, mouse/trackpad** to **click** and **drag**). -Pupils know how to use a **mouse/trackpad** in different ways (open a **program** and create a picture).	**Information Technology Around Us** (Networks and computing systems) -Pupils know the uses of **information technology** in school and beyond. -Pupils know how **information technology** helps us. -Pupils know how to use information technology **safely.** -Pupils know that choices are made when using **information technology.**	**Connecting Computers** (Networks and computing systems) - -Pupils know how **digital devices** function, using **input** and **output.** - Pupils know the physical components of a **network** and that a **computer network** is made up of multiple **devices.** -Pupils know how **digital devices** can change the way we work. -Pupils know how a **computer network**	**The internet** (Networks and safety and security) -Pupils know how **networks** physically connect to other networks. -Pupils know how **networked devices** make up the **internet.** -Pupils know how **websites** can be shared via the **World Wide Web (WWW).** -Pupils know how content can be added and accessed on the **World Wide Web (WWW).**	**Sharing Information** (Networks and effective use of tools) -Pupils know and can explain that computers can be connected together to form **systems,** and that these feature **inputs, outputs** and **processes.** -Pupils know the role of **computer systems** in our lives. -Pupils know how to use **search engines.** -Pupils know how **search engines** select results.	**Internet Communication** (Networks and effective use of tools) -Pupils know the importance of **internet addresses** and know how these are used to access **websites.** -Pupils know how **data** is **transferred** across the **internet.** -Pupils know how sharing information **online** can help people to work together and can evaluate different

		-Pupils know how to use a **keyboard** to type on a **computer** and move the **cursor** and **delete** letters. -Pupils know how to use the **keyboard** to edit text. -Pupils know how to use **technology** responsibly.		can be used to share information. -Pupils know how **digital devices** can be connected.	-Pupils know how the content of the **WWW** is created by people. -Pupils know that there are rules to protect content on the **WWW.** -Pupils know that not everything on the **WWW** is true, and why it may not be honest, accurate or legal.	-Pupils know how search results are ranked and recognise why the order of results is important, and to whom.	ways of working together **online.** -Pupils know how we communicate using **technology** and can evaluate different methods of **online** communication. -Pupils know how to access shared **files** stored **online.**
Autumn 2 **Data and Information**	**Grouping Data** -Pupils know that things can be **grouped, comparing** and spotting similarities and differences, beginning to work out **rules**. -Pupils know that objects can be **labelled.**	**Grouping Data** (Data and information and algorithms) -Pupils know objects can be **labelled** -Pupils know that objects can be counted. -Pupils know that we can describe objects in different ways. -Pupils know they can count objects with the same **properties.** -Pupils know they can **compare** groups of objects. -Pupils know they can **group** objects and answer questions about these groups.	**Pictograms** (Data and information and effective use of tools) -Pupils know how to describe the **properties** of an object, count and compare them. Pupils know how to count and **compare** objects using **tally charts.** -Pupils know that objects can be **represented** as pictures. -Pupils know how to create a **pictogram.** -Pupils know how to select objects by **attribute** and make **comparisons.** -Pupils know that people can be described by **attributes.** -Pupils know how to explain that we can **present information** using a **computer.**	**Branching Databases** (Data and information and effective use of tools) -Pupils know how to create questions with yes/no answers. -Pupils know how to identify the **attributes** needed to **collect data** about an object. -Pupils know how to create a **branching database.** -Pupils know how to explain why it is helpful for a **database** to be well structured. -Pupils know how to plan the structure of a **branching database.** -Pupils know how to independently create an identification tool.	**Data logging** (Computer systems and data and information) -Pupils know that **data** gathered over time can be used to answer questions. -Pupils know how to use a **digital device** to collect **data** automatically. -Pupils know that a **data logger** collects **'data points'** from **sensors** over time. -Pupils know how to view and sort data. -Pupils know how a computer can help us **analyse data.** -Pupils know how to identify the **data** needed to answer questions. -Pupils know how to use data from **sensors** to answer questions.	**Flat-file database** (Data and information and effective use of tools) -Pupils know how to use a form to record information. -Pupils know how to compare paper and **computer-based databases.** -Pupils know how to navigate a **flat fil database** to compare information -Pupils know how you can answer questions by grouping and then sorting **data.** -Pupils know how **'and** and **or'** can be used to **refine data.** Pupils know how **filters** can **refine data** and **charts** -Pupils know that tools can be used to select specific **data.** -Pupils know that **computer programs** can be used to	**Introduction to Spreadsheets** (effective use of tools and data and information) -Pupils know how to create a **data set** in a **spreadsheet.** -Pupils know the inputs and outputs in a spreadsheet. -Pupils know how to build a **data set** in a **spreadsheet.** -Pupils know that **formulas** and operations can be used to **calculated data.** -Pupils know how to apply **formulas** to **data.** -Pupils know how to create a **spreadsheet** to plan an event. -Pupils know to choose suitable ways to **present data.**

						compare data visually. -Pupils know how to use a **real-world database** to answer questions.	
Spring 1 **Programming A**	**Programming** -Pupils know that things can be **grouped, comparing** and spotting similarities and differences, beginning to work out **rules**. -Pupils know that objects can be **labelled.** -	**Moving a Robot** (Algorithms and programming) -Pupils know what a given **command** will do, predicting and matching it to an **outcome**. -Pupils know how to combine forwards and backwards **commands** to predict and make a **sequence.** -Pupils know how to use left and right **commands** to move a **robot**. -Pupils know how to combine four direction **commands** to make **sequences.** -To know how to plan a simple **program** and **debug** the **program**, knowing what it should do. -Pupils know how to find more than one solution to a problem.	**Robot algorithms** (Algorithms and programming) -Pupils know how to describe a series of **instructions** to create a **sequence.** -Pupils know how to explain what happens when we change the order of instructions. -Pupils know how to **program** and predict a sequence for a **floor robot.** -Pupils know that programming projects can have **code** and artwork -Pupils know how to design an **algorithm.** -Pupils know how to create and **debug** a program that has been created.	**Sequencing Sounds** (Programming and design and development) -Pupils know how to use the programming environment of **Scratch**. -Pupils know that commands in **Scratch** are represented in **blocks** have an outcome. -Pupils know how to create a **program** using a design and **sequence.** -Pupils know that sprites are controlled by **commands.** -Pupils know that a **sequence** of **commands** can have an order. -Pupils know how to create a project from a task description using **sprites** controlled by movement **commands** and sound **commands**. - Pupils know how to change the appearance of a project.	**Repetition in Shapes** (Algorithms and programming) -Pupils know that accuracy in **programming** is important. -Pupils know how to create a **program** on a computer by typing in **commands** in a **text-based** language. -Pupils know how to write an **algorithm** to achieve an outcome. -Pupils know what **‘repeat’** means. -Pupils know how to write a **code** and change the **value** of a **command.** - Pupils know how to use and modify a **count-controlled loop, knowing which values to change** to produce a given outcome. -Pupils know how to decompose a task into small steps. -Pupils know how to use a procedure in a program and develop the program by **debugging** it.	**Selection in Physical Computing** (Programming and computing systems) -Pupils know how to control a simple **circuit** and connect it to a **microcontroller controlling** an **LED.** -Pupils know what an **infinite loop** does. -Pupils know how to connect more than one **output device** to a **microcontroller.** -Pupils know how to write a **program** that includes **count-controlled loops** and that these control the output**.** -Pupils know that a **loop** can stop when a **condition** is met. - Pupils know that a **loop** can be used to repeatedly check whether a **condition** has been met and that if it has can start an action. -Pupils know that a **condition (if, then)** can control a **program**. -Pupils know how to design a physical project that includes **selection.**	**Variables in Games** (Programming and design and development) -Pupils know that a **variable** is used in a program and these can hold a number of letters, knowing that they have names and **values.** -Pupils know that **program variables** can hold the place of a single **variable**. -Pupils know that events in a **program** can set **variables.** -Pupils know how to improve a game by using **variables.** -Pupils know how to design a project that builds on a given example. -Pupils know how to use a design to create a project. -Pupils know how to evaluate their project.

						-Pupils know how to create a **program** that controls a **physical computing project.** -Pupils know how to test and **debug** the project.	
Spring 2 **Creating Media (Graphics)**	**Creating media** -Pupils know to make simple marks on a **device**, using a **painting tool**.	**Digital Painting** (Effective use of tools and creating media) -Pupils know what the different freehand **paint tools** do. -Pupils know how to use the **shape tool** and the **line tool.** -Pupils know how to change the colour and brush sizes. -Pupils knows how to make careful choices when painting a **digital picture** in the style of an artist. -Pupils know which tools to choose and use. -Pupils know how to use a **computer** to paint a picture. -Pupils know how to compare painting a picture on a **computer** and on paper.	**Digital Photography** (Effective use of tools and creating media) -Pupils know which **digital device** to use to capture a **digital phot0.** -Pupils know how to use a **digital device** to take a **photograph** in either portrait or landscape. -Pupils know how to make choices when taking a **photograph.** -Pupils know what makes a good **photograph.** -Pupils know how **photographs** can be improved. -Pupils know how to use **tools** to change an **image** -Pupils know that **photos** can be changed				**Web Page Creation** (Creating media and design and development) Pupils know the different types of **media** used on **websites**. -Pupils know that websites are written in **HTML**. -Pupils know how to review an existing **website** and consider its structure. -Pupils know the common features of a **web page.** -Pupils know about the **ownership** and use of images **(copyright)**. -Pupils know the term fair use and can find **copyright free images.** -Pupils know how to add content to a **webpage** and preview it. -Pupils know that there is a need for a **navigation path** and can link **webpages** using **hyperlinks.** -Pupils know the implications of linking to content

							owned by other people.
Spring 2 **Creating Media (Video)**	**Creating media** -Pupils know how to take **photographs** using a **device**.			**Stop-frame Animation** (Effective use of tools and creating media) -Pupils know that **animation** is a sequence of drawings or photographs. -Pupils know how to relate **animated movement** with a sequence of images. -Pupils know how to plan an **animation.** -Pupils know how to create a flip-book style **animation,** explaining how it works. -Pupils know how to work consistently and carefully. -Pupils know how to review and improve an **animation.** -Pupils know how to evaluate the impact of adding other **media** to an **animation.**		**Video Production** (Effective use of tools and creating media) -Pupils know what makes a video effective. -Pupils know which digital devices can record video. -Pupils know the features on a **digital video recording device,** including a **microphone.** -Pupils know how to capture **video** using a range of techniques. -Pupils know how to create a storyboard. -Pupils know that **video** can be improved through **reshooting** and **editing** and know which **tools** to use. -Pupils know the impact of the choices made when making and sharing a video. -Pupils know how to save, retrieve and export video content.	
Spring 2 **Creating Media (Audio)**					**Audio Production** (Effective use of tools and creating media) -Pupils know that **sound** can be **recorded**. -Pupils know that **input** and **output** devices are used to		

					record and **play sound**. -Pupils know how to use a **computer** to **record audio**. -Pupils know that **audio recordings** can be **edited/trimmed** and **saved** as an **editable document**. -Pupils know how to combine **audio** to enhance their **podcast** project. -Pupils know how to evaluate the effective use of **audio**.		
Summer 1 **Programming B**	**Programming** Pupils know how to **order** and **sequence,** including for stories. -Pupils know that problems can be broken down in to **steps.**	**Programming Animations** (Programming and design and development) -Pupils know how to choose a **command** for a given purpose (move a **sprite**). -Pupils know that a series of commands (**blocks**) can be joined together to create an **algorithm**. -Pupils know how to **run** and **test** a **program** using a **start** block. -Pupils know the effect of changing a **value** of a **block.** -Pupils know that each **sprite** has its own instructions and how to add and **delete** more than 1 **sprite**.	**Programming Quizzes** (Programming and design and development) -Pupils know that a sequence of **commands** has a start and know how to run a **program**. -Pupils know that a sequence of **commands** has an outcome and the outcome can be changed. -Pupils know how to create a **program** using a given design, creating **algorithms** for a **sprite’s** actions using a sequence of blocks. -Pupils know how to change a given design. -Pupils know how to create a **program**	**Events and Actions in Programs** (Programming and design and development) **-**Pupils know how a **sprite** moves in an existing project using **blocks.** -Pupils know how to **program** movement using a sequence of **commands** and create a program to move a **sprite** in four directions. -Pupils know how to adapt a **program** to a new context using program extensions and **blocks**. -Pupils know how to develop a **program** by adding features from a given set of **blocks** and choose suitable **keys** to turn on additional features.	**Repetition in Games** (Programming and design and development) -Pupils know how to use **count-controlled** and **infinite loops** in a different programming environment. -Pupils know that in programming there are infinite loops and count-controlled loops. -Pupils know how to develop a design that includes two or more loops which run at the same time and know that more than one **process** can **run** at once. -Pupils know how to modify an **infinite loop** in a given **program**.	**Selection in Quizzes** (Algorithms and programming) -Pupils know how **selection** is used in **computer programs** and how **conditions** are used. -Pupils know that a conditional statement connects a **condition** to an outcome. -Pupils know how to use **selection** in an **infinite loop** to check a **condition.** -Pupils know the **condition** and outcomes in an ‘if… then… else…’ statement. -Pupils know how **selection** directs the flow of a **program** and it can branch	**Sensing Movement** (Programming and computing systems) -Pupils know how to create a **program** to run on a **controllable device** and test the **program** on an **emulator.** -Pupils know that **programs** can be **transferred** to a **controllable device.** -Pupils know that **if, then** and **else statements** can be used to control the flow of a **program**, knowing the importance of the order of these. -Pupils know how to update a **variable** with a user **input** and that **conditions** can be used to change **variables**.

		-Pupils know how to design the parts of a project. -Pupils know how to use algorithms to create a program and test what has been created.	using their own design. -Pupils know how improve their project by adding features and **debugging**.	-Pupils know how to fix **bugs** in a **program**. -Pupils know how to design and create a maze-based challenge.	-Pupils know how to design a **project** that includes repetition. -Pupils know how to evaluate the steps they followed when building their **project**. -Pupils know how to use existing **code** on new **sprites.**	according to a **condition**. -Pupils know how to design a **program** that uses **selection** and identify the outcome of user input in an **algorithm.** -Pupils know how to create a **program** that uses selection and implement an **algorithm** to create the first section of my **program.** -Pupils know how to evaluate their **program.**	-Pupils know how to use a **conditional statement** to compare a **variable** to a value and use **operands (<>=)** in an if, then statement. -Pupils know how to design a project that uses **inputs** and **outputs** on a **controllable device.** -Pupils know how to develop a **program** to use **inputs** and **outputs** on a **controllable device** and use a range of approaches to find and **fix bugs.**
Summer 2 **Creating Media (Text)**	**Creating media** -Pupils know some letters on a **keyboard**.	**Digital Writing** (Effective use of tools and creating media) -Pupils know how use a **computer** to write using a **word processing**. -Pupils know and find **keys (space, back space, letters and numbers)** on a **keyboard.** -Pupils know how to **add** and **remove** text on a computer. Pupils know that the look of **text** can be changed on a **computer.** -Pupils know how to type a capital letter using **caps lock.** -Pupils know what the **tool bar** is and how to use **bold**, **italic**, and **underline.**		**Desktop Publishing** (Effective use of tools and creating media) -Pupils know how **text** and **images** convey information. -Pupils know the difference between **text** and **images** and the advantages and disadvantages of using **text** and **images**. -Pupils know that **text** and **layout** can be **edited** (**font style, size, and colours** for a given purpose). -Pupils know how to choose the appropriate page settings (**templates, ‘page orientation,’ placeholders**). -Pupils know how to add content to a			

		-Pupils know how to make careful choices when changing **text** by **double-clicking** and dragging to select. -Pupils know how to change the **font**. -Pupils know how to improve their writing and can explain why they used the **tools** that they chose. Pupils know how to use **'Undo'** to remove changes. -Pupils know the similarities and difference between writing on paper and on a **computer** and can say why they prefer typing or writing.		**desktop publishing publication**. -Pupils know how different layouts can suit different purposes and choose a suitable layout for a given purpose. -Pupils know the benefits of **desktop publishing** and can compare work made on **desktop publishing** to work created by hand.			
Summer 2 **Creating Media (Graphics)**					**Photo Editing** (Effective use of tools and creating media) -Pupils know that the composition of **digital images** can be changed by **rotating**, **cropping** an **image** and **changing colours**. **-Pupils know** how **cloning** and **colour effects** can be used in photo editing. -Pupils know that images can be **combined** using **tools** to **select** and **copy** part of an image.	**Introduction to Vector Graphics** (Effective use of tools and creating media) **-Pupils know** that **drawing tools** can be used to produce different outcomes and that **vector drawings** are made using shapes or **objects**. -Pupils know how to create a **vector drawing** by combining **shapes, resizing, rotating and duplicating objects.** -Pupils know which **tools** to use to	**3D Modelling** (Effective use of tools and creating media) -Pupils know that you can work in three dimensions on a **computer** and add 3D shapes to a project. -Pupils can move 3D shapes relative to one another. -Pupils know that digital 3D objects can be modified by **resizing**,

					-Pupils know how to **combine** images for a purpose. -Pupils know how changes can improve an image.	achieve a desired effect and improve consistency including **zoom, alignment grids** and **resize handles.** -Pupils recognise that **vector drawings** consist of layers and how to add a new **layer**, change the **order** of **layers**. -Pupils know how to **group** and **ungroup objects** to make them easier to work with by **copying** part of a drawing and **duplicating** several objects **-Pupils know how to** apply what they have learned about **vector drawings** to create a **vector drawing** for a specific purpose and reflect on the skills they have used and why they have used them.	**lifting, lowering** and **recolouring** a 3D object. -Pupils know that objects and models can be **combined, grouped, rotated** and **duplicated** in a 3D model. -Pupils know how to create a 3D model for a given purpose. -Pupils know how to accurately **size** the 3D objects and show that **placeholders** can create holes in 3D objects. -Pupils know how to plan and create their own 3D model and explain how it could be improved and modified.
<u>Summer 2</u> Creating Media (Audio)	**Creating media** Pupils know how to **record sounds** and speech using a **microphone** and **device**.		**Making digital music** (Creating media and design and development) -Pupils know how music can make us feel and identify simple differences in pieces of music. -Pupils know that there are patterns in music and can follow				

			and create a rhythm patterns. -Pupils know how to change **pitch** using a **computer** and using images to create sounds. -Pupils know how to use a **computer** to create a **musical pattern** using sequences of notes. -Pupils know how create music/rhythm for a purpose. -Pupils know how to review and refine their **computer** work and explain how they changed it.				