Curriculum Aims and Overview Computing

The study of computing is a necessity in our ever- changing world where digital technology is improving at an increasing rate.
Technologies such as artificial intelligence, automation and robotics are changing the way that we live, work and socialise. We recognise the vital role that we as educators play in teaching young people the skills they'll need to thrive in a digital future, through the national computing curriculum. The computing curriculum offers powerful benefits for young people. Early use of digital technology improves children's language skills and promotes social development and creativity. Having a deeper understanding of computing also helps students to be better equipped when tackling maths, science and engineering problems in STEM classes.
Socially, the computing curriculum offers a step-up to the 20% of students in the UK who don't have access to an internet-connected device at home for learning, and the 10% of UK households who still do not have internet access. Digital and computing skills are more important now than ever. Digital skills are a universal requirement in the job market and data developing digital skills makes career progression more likely, unlocks more opportunities and ultimately increases social mobility.

Scope and Sequencing
Computing has deep links with mathematics, science and design and technology, and provides insights into both natural and artificial systems. The core of computing is computer science, in which pupils are taught the principles of information and computation, how digital systems work and how to put this knowledge to use through programming. Building on this knowledge and understanding, pupils are equipped to use information technology to create programs, systems and a range of content. Computing also ensures that pupils become digitally literate – able to use, and express themselves and develop their ideas through, information and communication technology – at a level suitable for the future workplace and as active participants in a digital world.
Our curriculum is based around the TEACH computing curriculum which has been developed by the National Centre for Computing Education funded by the DfE. It is built around an innovative progression framework where computing content has been organised into interconnected networks, created by subject experts, using the latest pedagogical research and teacher feedback.

There are four core pillars underpinning the discipline of computing

1. Computer networks and systems
2. Creating media
3. Programming
4. Data and information

Identifying and combining these core strands works towards the overall goal of children being able to use their substantive knowledge to create their own content.

Substantive and Disciplinary Content in Computing
Every subject is unique and includes its own substantive content and disciplinary content. The INSPIRE computing curriculum is designed to ensure that pupils not only have broad and strong substantive knowledge but also understanding of the discipline of computing. Pupils learn both language 'facts' and how to make sense of them simultaneously.
Disciplinary knowledge in computing is the use and interpretation of substantive knowledge in order to develop original digital content and programs. The core strands are Data and Information, Creating Media, Computing Systems and Networks and Programming. We also focus heavily on E-safety and keeping children safe in our digital world.

Substantive Knowledge
Computing is a cumulative discipline. Pupils' knowledge of what we often call substantive concepts such as creating media come up time and time again in the curriculum. We know if pupils are able to build up knowledge of these concepts, building richer and richer schemata of these concepts and terms over time, it can help them access increasingly complex material throughout the curriculum, which helps them to learn, understand and remember more – meaning they make more progress. The curriculum is sequenced to allow children to build on prior knowledge

Disciplinary Knowledge
Learning computing involves the development of both substantive knowledge (the 'stuff' of computing) and familiarity with the 'second-order' or procedural disciplinary concepts, that shape the way in which the 'stuff' or 'substance' is understood, organised and debated, as well as the ways in which it is actually generated.
The simplest way to think about the difference is – the substantive knowledge is the **'what**' and the disciplinary knowledge is the **'how'.**
As we're teaching we need to interweave the **what** and the **how** for our children. Thinking linguistically is vital. Simply 'knowing' the steps in programming for example is not computational thinking. The best we could say is that it *enables computational* thinking. We need facts in order to think, but we also need concepts to enable us to group bits of information, or facts, together.

Core Concepts:

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
EYFS	One and Only Me	Celebrations	Traditional Tales	Local area/ London contrast and compare/ Spring/ Easter	Under The Ground	Windrush
Y1	Computing systems and networks – Technology around us https://docs.google.com/forms/technologyaroundus	Creating media – Digital painting https://docs.google.com/forms/digitalpainting	Programming A – Moving a robot https://docs.google.com/forms/movingarobot	Creating media – Digital writing https://docs.google.com/forms/digitalwriting	Creating media – Digital writing – *Application unit skills from Spring 2*	Programming B – Programming animations https://docs.google.com/forms/programminganimations
Y2	Computing systems and networks – IT around us https://docs.google.com/forms/ITaroundus	Creating media – Digital photography https://docs.google.com/forms/digitalphotography	Programming A – Robot algorithms https://docs.google.com/forms/robotalgorithms	Data and information – Pictograms https://docs.google.com/forms/pictograms	Creating media – Digital writing (Google slides)	Programming B – Programming quizzes https://docs.google.com/forms/quizzes
Y3	Computing systems and networks – Connecting computers https://docs.google.com/forms/connectingcomputers	Creating media – Animation https://docs.google.com/forms/creatingmedia	Programming A – Sequencing sounds https://docs.google.com/forms/sounds	Data and information – Branching databases https://docs.google.com/forms/branchingdatabases	Creating media – Desktop publishing https://docs.google.com/forms/desktoppublishing	Programming B – Events and actions https://docs.google.com/forms/eventsandactions
Y4	Computing systems and networks – The internet https://docs.google.com/forms/theinternet	Creating media – Audio editing https://docs.google.com/forms/audioediting	Programming A – Repetition in shapes https://docs.google.com/forms/repetitioninshapes	Data and information – Data logging https://docs.google.com/forms/datalogging	Creating media – Photo editing https://docs.google.com/forms/photoediting	Programming B – Repetition in games https://docs.google.com/forms/repetitioningames
Y5	Computing systems and networks – Sharing information https://docs.google.com/forms/sharinginformation	Programming A – Selection in physical computing (Crumble) https://docs.google.com/forms/selectioninphysicalcomputing	Creating media – Video editing https://docs.google.com/forms/videoediting	Data and information – Flat-file databases https://docs.google.com/forms/flatfiledatabases	Creating media – Vector drawing https://docs.google.com/forms/vectors	Programming B – Selection in quizzes https://docs.google.com/forms/selectioninquizzes
Y6	Computing systems and networks – Communication https://docs.google.com/forms/communication	Creating media – Web page creation https://docs.google.com/forms/webpagecreation	Programming B – Sensing (Micro:Bit) https://docs.google.com/forms/sensing	Data and information – Spreadsheets https://docs.google.com/forms/spreadsheets	Creating media – 3D modelling https://docs.google.com/forms/3dmodelling	Programming A – Variables in games https://docs.google.com/forms/variablesingames

THE BIG IDEAS- SUBSTANTIVE CONCEPTS		
S.C. 1 – Computing systems and networks	**S.C. 2 – Creating media**	**S. C. 3 – Data and information**
S.C. 4 – Programming	**S.C. 5 – E-safety** This runs across every unit	

Whole School Overview

COMPUTING PROGRESSION GRID

The **National Centre for Computing Education (NCCE)** is funded by the Department for Education and supporting partners, and marks a significant investment in improving the provision of computing education in England.

TEACH computing curriculum has been developed as part of this in line with our work with the Computing Hub.

Computing Progression Grid Document			
EYFS	**Three & Four year olds**	**Reception**	**ELG**
Personal, Social and Emotional Development	Increasingly follow rules, understanding why they are important. (Understand why we have rules when using technological equipment and obey these to stay safe).	Show resilience and perseverance in the face of a challenge when using technology that they may find difficult.	Be confident to try new activities and show independence and perseverance in the face of challenge.
Physical Development	Matching their developing physical skills to tasks and activities in the setting.	Develop their small motor skills so that they can use a range of tools competently, safely and confidently.	
Understanding the World	Explore how things work (e.g. use a remote-control car, make basic movements on a whiteboard, turn on and off different devices.		
Expressive Arts and Design		Explore, use and refine a variety of artistic effects to express their ideas and feelings (e.g. using technology to record their work such as photos).	Safely use and explore a variety of materials, tools and techniques, experimenting with colour, design, texture, form and function. (e.g. drawing digital pictures using interactive whiteboards and Ipads)

Early Learning Goals:	**Digital Literacy**	**Information Technology**	**Computer science**	**Vocabulary**
Children recognise that a range of technology is used in places such as homes and schools. **They select and use technology for particular purposes.** **Children find out about and use a range of everyday technology.** **They select appropriate applications that support an identified need – for example in deciding how best to make a record of a special event in their lives, such as a journey on a steam train.**	**I can identify what technology is in the classroom.** **I can identify and explain, the uses of technology, in and around, my classroom (including Twitter etc.)** **I can discuss what technology is in my home and what is used for.** **I can explain that info**	**I can turn on digital equipment.** **I can handle technology with care. I can interact with technology.** **I can turn on/off digital equipment.** **I can interact with technology purposefully (navigating an iPad).** **I can use technology to take a picture.** **I can use technology to record a video**	**I can complete a simple programming sequence using a range of technology (BeeBots, programming games online)** **I can give instructions using Walkie Talkies (algorithms)**	**Sequence** **Technology** **Digital Equipment** **information** **Tweet** **program**

PROGRESSION - SUBSTANTIVE KNOWLEDGE					
AREA OF STUDY: Computing Systems & Networks					
YEAR 1	**YEAR 2**	**YEAR 3**	**YEAR 4**	**YEAR 5**	**YEAR 6**
Identify technology in the classroom and how it helps us (DL) Name the main parts of a computer (IT) Switch on and log into a computer (IT) Use a mouse to click and drag (IT) Use a mouse to create a picture (IT) Use a mouse to open a program (IT) Save work to a file (IT) Type their name on a keyboard (IT) Delete letters (IT) Open work from a file (IT) Use the arrow keys to move a cursor (IT) Identify rules to keep us safe and healthy when using technology (DL) Lesson planning	Describe the uses of computers (DL) Identify examples of computers and understand that a computer is part of IT (DL) Identify examples of IT in school and how we use it (DL) Identify examples of IT beyond school and how we use it (DL) Explain how IT helps us (DL) Identify rules for how to use IT safely (DL) Use the correct IT for different types of activities (IT) Lesson planning	Explain that digital devices accept inputs and produce outputs (CS) Identify input and output devices (CS) Explain how we use digital devices for different activities (DL) Understand the similarities and differences between digital and non-digital tools (DL) Discuss why we need a network switch (CS) Explain how messages are passed through different connections (CS) Demonstrate how information can be passed between devices (CS) Explain the role of a switch, server and wireless network point in a network (CS) Identify how devices in a network are connected together (CS) Identify the benefits of computer networks (DL) Lesson planning	Demonstrate how information is shared across the internet (CS) Discuss why a network would need protecting (DL) Recognise how networked devices make up the internet (CS) Describe how to access websites on the WWW and where they are stored when uploaded to the WWW (IT) Explain what media can be found on websites (DL) Recognise that we can add content to the WWW (DL) Recognise how the content of the WWW is created by people (CS) Explain that not everything on the WWW is true (DL) Explain why we should think carefully before sharing or resharing content (DL) Explain why some information we find online might not be honest, accurate or legal (DL) Lesson planning	Compare results from different search engines (IT) Complete a web search to find specific information (IT) Refine web searches (IT) Recognise the role of web crawlers in creating an index (CS) Relate a search term to the search engine's index (CS) Explain how search engines are ranked (CS) Explain why the order of results is important and to whom (CS) Choose methods of communication to suit particular purposes (IT) Compare different methods of communication over the internet (DL) Decide what I should and should not share (DL) Understand that what they share may not be private (DL) Lesson planning	Describe that a computer system features inputs, processes and outputs (CS) Explain that computers are connected together to form systems (CS) Explain the benefits of a computer system (DL) Identify tasks that are managed by computer systems and the role a human plays in this (CS) Explain that data is transferred over networks in packets (CS) Explain that networked digital devices have unique addresses (CS) Recognise that connected digital devices can allow us to access shared files stored online (CS) Send information over the internet in different ways (IT) Contribute to a shared project online (IT) Explain how the internet enables effective collaboration (DL) Lesson planning
Disciplinary focus: **Core: Logic** **Sub: Decomposition**	**Disciplinary focus:** **Core: Pattern** **Sub: Evaluation**	**Disciplinary focus:** **Core: Decomposition** **Sub: Abstraction**	**Disciplinary focus:** **Core: Decomposition** **Sub: Logic**	**Disciplinary focus:** **Core: Abstraction** **Sub: Patterns**	**Disciplinary focus:** **Core: Decomposition** **Sub: Evaluation**
SMSC question: **Is the use of computers healthy?**	**SMSC question:** **Do computers always help us?**	**SMSC Question:** **Do computer networks always improve communication?**	**SMSC Question:** **How can the internet cause problems in society?**	**SMSC Question:** **How could bias be developed through search engines?**	**SMSC Question:** **What are the key problems with sharing personal information on the web?**
VOCABULARY- CORE vocabulary highlighted yellow					
Technology **Computer** **Mouse** **Trackpad** **Keyboard Screen** **Double-click** **Typing**	**Information technology** **Barcode** **Scanner** **Scan** **Devices**	**Digital device** **Input** **Process** **Output** **Network** **Program** **Digital/Non-digital** **Connection Network** **Network switch** **Server** **Wireless access point** **Network cable** **Network socket**	**Internet Router** **Network security** **Website/Webpage** **Web address** **Web browser** **World wide web** **Files** **Download** **Sharing** **Ownership** **Permission** **Adverts**	**System** **Digital Search** **Search engine** **Refine** **Index** **Web Crawler** **Bot** **Ordering/Ranking** **Search engine optimisation**	**Communication** **Protocol** **Data** **Internet protocol (IP) address** **Domain name** **Packet Header** **Data payload** **Collaboration**

PROGRESSION - SUBSTANTIVE KNOWLEDGE					
AREA OF STUDY: Creating Media					
YEAR 1	**YEAR 2**	**YEAR 3**	**YEAR 4**	**YEAR 5**	**YEAR 6**
Draw lines and marks on a screen (IT) Use paint tools to create a picture (IT) Use the shape and line tool to make marks (IT) Use the shape and line tools to create a picture (IT) Make appropriate shape and colour choices when painting a digital picture (IT) Choose the best paint tool for the purpose (IT) Change the brush size and colour (IT) Explain the differences between painting on a computer and painting online (DL) Lesson planning Identify and find keys on a keyboard (IT) Open a word processor (IT) Enter text into a computer (IT) Use backspace to remove text (IT) Use the letter, number and space keys (IT) Identify the toolbar (IT) Use bold, italic and underline (IT) Type in capital letters (IT) Change the font (IT) Select text by clicking and dragging (IT) Select a word by double clicking (IT) Use the undo tool (IT) Make changes to text on a computer (IT) Lesson planning	Use a digital device to take a photograph (IT) Take photos in portrait and landscape formats (IT) Improve a photo by retaking it (IT) Explore the effect light has on a photograph (IT) Use tools to edit an image (IT) Identify photos that have been changed (DL) Lesson planning Identify the uses and purpose of google slides Open a blank slides document Identify the text features including text boxes Type, delete and manipulate text within a text box Change colour and font of text Use different text box types to add headings to blank slides Identify how to add and delete slides Add pictures into a slides document Add shapes into a slides document Adjust layout, colour and background within different slides Use the master slide to keep your slides consistent Use the collaboration features to share and edit slides presentations Add animations to slides Add transitions to slides Design a presentation around a chosen theme Present your final presentation Lesson planning	Explain that animation is a sequence of drawings or photographs (IT) Create an effective stop-frame animation (IT) Plan an animation (IT) Use onion-skinning to help make small changes between frames (IT) Review and improve their animation (IT) Add additional media to their animation (IT) Lesson planning Explain the differences between text and images (IT) Change font style, size and colours for a given purpose (IT) Create a template (IT) Change the page from portrait to landscape (IT) Recognise and understand the importance of placeholders (IT) Edit text to communicate more clearly (IT) Copy and paste text and images into a document (IT) Choose a suitable layout for a given purpose (IT) Identify the uses of desktop publishing in the real world (DL) Lesson planning	Identify digital devices that can record sound and play it back (DL) Identify inputs and outputs required to play or record sound (CS) Use a digital device to record sound (IT) Save a digital recording as a file (IT) Open a digital recording from a file (IT) Edit sections of a recording (IT) Use editing tools to arrange sections of audio (IT) Lesson planning Identify changes we can make to an image (DL) Explore how images can be changed in real life (DL) Change the composition of an image (IT) Choose effects to make an image fit a scenario (IT) Choose appropriate tools to retouch an image (IT) Combine parts of images to create new images (IT) Recognise that not all images are real and explain how they know (DL) Compare original images with edited ones (DL) Lesson planning	Explain that a video is a visual media format (CS) Identify digital devices that can record video (DL) Experiment with different camera angles (IT) Use a microphone (IT) Capture a video using a range of filming techniques (IT) Save video content (IT) Edit by reshooting (IT) Store, retrieve and export video to a computer (IT) Share a video (IT) Lesson planning Understand that vector drawings are made using shapes (IT) Identify the main drawing tools (IT) Create a vector drawing by combining shapes (IT) Move, resize and rotate objects (IT) Duplicate objects (IT) Use alignment grids and resize handles (IT) Modify objects to create different effects (IT) Use the zoom tool to add more detail (IT) Change the order of layers in a vector drawing (IT) Identify which objects are in the front layer or back layer of a drawing (IT) Group objects (IT) Evaluate vector drawings (IT) Lesson planning	Review and explore websites (DL) Recognise common features of a web page (IT) Find and understand the importance of copyright-free images (DL) Add content to their own web page (IT) Preview their own web page (IT) Create multiple web pages (IT) Create hyperlinks (IT) Link web pages using hyperlinks (IT) Lesson planning Create 3D digital objects (IT) Manipulate (select/ move/ delete) 3D digital objects (IT) Change the colour of 3D objects (IT) Resize 3D objects (IT) Rotate 3D objects (IT) Select and duplicate 3D objects (IT) Group digital 3D shapes and a placeholder to create a hole in an object (IT) Modify multiple 3D objects in a variety of ways (IT) Create a 3D digital model using a variety of 3D shapes (IT) Lesson planning
Disciplinary Focus **Core: Abstraction** **Sub: Decomposition**	**Disciplinary Focus** **Core: Evaluation** **Sub: Patterns**	**Disciplinary Focus** **Core: Decomposition** **Sub: Abstraction**	**Disciplinary Focus** **Core: Abstraction** **Sub: Evaluation**	**Disciplinary Focus (Video Production)** **Core: Evaluation**	**Disciplinary Focus** **Core: Decomposition** **Sub: Evaluation**

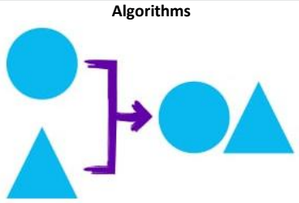
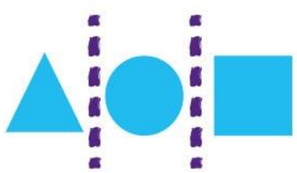
				Sub: Patterns **Disciplinary Focus (Vector Drawings)** **Core: Decomposition** **Sub: Evaluation**	
SMSC question: **Is it honest to change photos?**	**SMSC question:** **Do we need cameras if we have phones?**	**SMSC question:** **Is there still a place for writing by hand in a world where computers can help us present information?**	**SMSC question:** **What impact might photo shopping have on self-esteem?**	**SMSC question:** **What problems might uploading videos cause?**	**SMSC question:** **Can companies without online presence compete with others?**
VOCABULARY- CORE vocabulary highlighted yellow					
Paint Tool Paintbrush Erase Fill Undo Shape Line Colour Style Pointillism Size Word processor Space Backspace Text cursor Caps lock Toolbar Bold Italic Underline Font Format Redo	Device Camera Photograph Capture Image Landscape Portrait Framing Compose Flash Focus Background Slides Font Size Effects Presentation Colour Animation Transition	Animation Flipbook Stop-frame animation Frame Sequence Image Onion skinning Evaluate Text Images Communicate Template Orientation Placeholder Layout Content Desktop publishing Copy Paste	Audio Microphone Speaker Headphones Sound Podcast Edit Trim Align Layer Import/Export Record Playback Crop Rotate Effects Retouch Clone Combine Background Foreground Zoom	Video Panning Close up Lens Mid-range Long shot High/Normal/Low angle Static camera Zoom Tilt Filming Vector Drawing tools Object Move Rotate Duplicate Resize Modify Layers Group/Ungroup	Website Webpage Browser Media HTML Header Copyright Fair use Home page Google site Preview Navigation path Hyperlink 3D Perspective Handles Lift Lower Duplicate Group Placeholder Construct

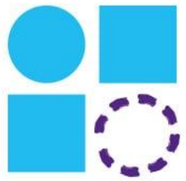
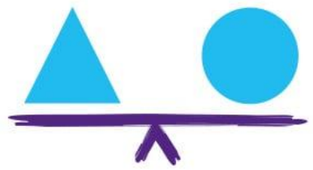
PROGRESSION - SUBSTANTIVE KNOWLEDGE					
AREA OF STUDY: Programming					
YEAR 1	**YEAR 2**	**YEAR 3**	**YEAR 4**	**YEAR 5**	**YEAR 6**
Match a command to an outcome (CS) Predict the outcome of a command (CS) Run a command on a device (CS) Predict the outcome of a sequence using forwards and backwards commands (CS) Start a sequence from the same place each time (CS) Predict the outcome of a sequence involving up to 4 commands (CS) Combine 4 direction commands to make a sequence (forwards, backwards, left and right) (CS) Debug their simple program (CS) Explain what their program should do (CS) Use 2 different programs to get to the same place (CS) Lesson planning Use more than 1 programming tool (CS) Use commands to move a sprite (CS) Run a program (CS) Use a start block in a program (CS) Use more than one block by joining them together (CS) Change the value of a block (CS) Add blocks to sprites (CS) Delete sprites (CS) Add more than 1 sprite to a project (CS) Create algorithms for sprites (CS) Test programs that they have created (CS) Lesson planning	Describe a series of instructions as a sequence (CS) Use the same commands to create algorithms for a range of sequences (CS) Use an algorithm to program a sequence on a floor robot (CS) Follow a sequence to predict an outcome (CS) Identify routes around a map (CS) Test a map to ensure it is usable (CS) Create an algorithm to meet a goal (CS) Use an algorithm to create a program (CS) Test and debug each part of a program (CS) Lesson planning Identify the start of a sequence and show how to run the program (CS) Change the outcome of a series of commands (CS) Match 2 sequences with the same outcome (CS) Predict the outcome of a sequence of commands (CS) Create a program using a given design (CS) Create a program using their own design (CS) Debug and improve their projects (CS) Lesson planning	Explore programming environments (e.g. Scratch) by identifying objects and commands (CS) Follow a design to create a program (CS) Create a sequence of connected commands (CS) Start programs in different ways (CS) Combine sound commands into a particular order (CS) Build a sequence of commands (CS) Make own design choices by assigning actions to sprites (CS) Implement their algorithm as code (CS) Create a project based on a task description (CS) Lesson planning Explain the relationship between an event and an action (CS) Program movement using 4 directions (CS) Use a programming extension (CS) Develop their program by adding different features (CS) Identify and fix bugs in a program against a given design (CS) Lesson planning	Program a computer by typing commands (CS) Write an algorithm in text-based language (CS) Use a count-controlled loop to produce a given outcome (CS) Modify a count-controlled loop to produce a given outcome (CS) Use a procedure in a program (CS) Design and create programs that include count-controlled loops (CS) Lesson planning Explore more than one programming environment (CS) Predict the outcome of snippets of code (CS) Know when to use infinite or count-controlled loops (CS) Run more than 1 process at a time (CS) Write programs that include 2 or more loops that run at the same time (CS) Re-use existing code snippets on new sprites (CS) Design programs that use repetition (CS) Create projects that include repetition (CS) Lesson planning	Create a simple circuit and connect to a microcontroller (CS) Connect more than 1 output component to a microcontroller (CS) Use count-controlled loops to control outputs (CS) Design a conditional loop (CS) Program a microcontroller to respond to an input (CS) Identify a condition and an action in a project (CS) Use selection to direct the flow of a program (CS) Design a physical project that includes selection (CS) Create a program (incl. testing and debugging) that includes a physical computing project (CS) Lesson planning Modify conditions in a program (CS) Create a program with different outcomes using selection (CS) Use selection in an infinite loop to check a condition (CS) Show that a condition can direct program flow in one of two ways (CS) Identify the outcome of user input in an algorithm (CS) Identify the setup code needed in their program (CS) Lesson planning	Know that a variable is something that is changeable (CS) Recognise that the value of a variable can be changed (CS) Use events in a program to set variables (CS) Create games that use variables (CS) Test and debug projects that include variables (CS) Lesson planning Test a program on an emulator (CS) Transfer programs to a controllable device (CS) Use selection to determine the flow of a program (CS) Use a variable in an 'if, then, else' statement to select the flow of a program (CS) Update a variable with a user input (CS) Use an operand (<>=) in an if, then statement Design a program that uses inputs and outputs on a controllable device (CS) Lesson planning

PROGRESSION OF SKILLS					
AREA OF STUDY: Data & Information					
YEAR 1	**YEAR 2**	**YEAR 3**	**YEAR 4**	**YEAR 5**	**YEAR 6**
	Enter data onto a computer (IT) Use a computer to view data in different formats (pictograms) Use a computer to create pictograms that arrange objects by an attribute (IT) Give simple examples of why information should not be shared (DL) Use a computer program to present information in different ways (IT) Lesson planning	Select attributes to separate objects and arrange into a tree structure (IT) Create a branching database (IT) Compare two branching database structures (IT) Identify objects using a branching database (IT) Compare two ways of presenting information (branching databases and pictograms) (IT) Lesson planning	Use a data logger to collect data Understand how it captures data over time Connect a data logger to a computer Use and interpret the data using intervals Select the information needed to solve a problem/ collect data based on a real-life problem Lesson planning	Understand what a database is Create a database on paper and using technology and compare Understand branching database Understand how to sort and group using a database Select questions to support solving a problem using a database Lesson planning	Understand what a spreadsheet is Understand the different elements of a spreadsheet (cell, column, formula) Understand how to enter data into a spreadsheet Be able to sort and find information in a spreadsheet Be able to use simple formulae within a spreadsheet Be able to format data Lesson planning
	Disciplinary Focus **Core: Evaluation** **Sub: Abstraction**	**Disciplinary Focus** **Core: Patterns** **Sub: Algorithms**	**Disciplinary Focus** **Core: Abstraction** **Sub: Algorithms**	**Disciplinary Focus** **Core: Abstraction** **Sub: Evaluation**	**Disciplinary Focus** **Core: Logic** **Sub: Evaluation**
	SMSC question: **Do we really need computers?**	**SMSC question:** **Do computers de-skill people?**	**SMSC question:** **Do computers contribute to a lack of family time?**	**SMSC question:** **By storing information online are we contributing to identity theft?**	**SMSC question:** **Is it ok for companies to sell information on their databases about customers?**
VOCABULARY- CORE vocabulary highlighted yellow					
	organise data Pictogram compare enter data tally chart objects block diagram object	Branching database attribute value structure information Attribute questions table compare	sensor data logger interval Analyse data set import table Input device export Analyse review	Database field criteria information record graph chart axis compare filter	Spreadsheet cells operation duplicate format Formula columns rows. common attribute sigma software

Disciplinary focus: **Core: Algorithms** **Sub: Logic**	**Disciplinary focus:** **Core: Algorithms** **Sub: Decomposition**	**Disciplinary focus:** **Core: Algorithms** **Sub: Logic**	**Disciplinary focus:** **Core: Algorithms** **Sub: Patterns**	**Disciplinary focus:** **Core: Algorithms** **Sub: Logic**	**Disciplinary focus:** **Core: Algorithms** **Sub: Evaluation**
SMSC Question: **How can we make sure we learn from our mistakes?**	**SMSC Question:** **Should we always follow instructions?**	**SMSC Question:** **Does it matter how much time we spend online? (screentime)**	**SMSC Question:** **How are mobile phones impacting on conversation?**	**SMSC Question:** **Is it better to play with friends online or in person?**	**SMSC Question:** **Is the gaming culture causing problems in society?**
VOCABULARY- CORE vocabulary highlighted yellow					
Command Instructions Directions Prediction Program Algorithm ScratchJr Bee-bot Sprite Block Value	Sequence Algorithm Order Route Debug Actions Bug	Scratch Programming Code Costume Backdrop Motion Sequence Event Design Extension block	Program Turtle Code snippet Pattern Repetition Count-controlled loop/Infinite loop Trace Value Decompose Procedure Animate Duplicate	Crumble Sparkle Microcontroller Components Connection Motor Repetition Switch LED Condition Selection Outcomes	Variable Micro:bit Make USB Condition Sensing Accelerometer Navigation Step counter

DISCIPLINARY SKILLS PROGRESSSION - COMPUTATIONAL THINKING

Concept	EYFS	KS1	KS2
Logic 	• Children start to reason about the world around them. • Children play with mechanical and electronic toys to start forming ideas about how they work. • Provide scenarios for children to predict and test. E.g. they might predict that big things sink and small things float. To test this, we might model trying different objects and then introduce a balloon and a stone.	• Use logical reasoning to predict the behaviour of simple programs, including both their own (perhaps for Scratch or a floor turtle) and other software (such as a game or a painting program).	• Explore a range of algorithms • use logical reasoning to think through each step to predict outcomes. • eg algorithms for mathematical calculations, for solving mazes or for looking up a dictionary definition. • children should be able to explain their predictions. • Debugging by thinking through each step and detecting where the errors are • Use logical reasoning to adjust the code.
Examples in other areas of the curriculum	English - children might use it to explain a character's actions in a story so far, and to predict what the character will do next. Science - children should be able to explain how they've arrived at certain conclusions from the results of experiments. History - children should understand how our knowledge is constructed from a variety of sources, and they should be able to discuss the logical connections between cause and effect. Design Technology - children need to reason what material is best suited to each part of a project.		
Algorithms 	• Teachers naturally create opportunities for **sequencing**, which is a key element of algorithms. • Children learn to take turns with others, to tidy up and line up. • Sequencing happens during roleplay activities; for example, the events which occur when we go to post a letter at the Post Office.	• There are many opportunities within the school day for children to understand algorithms and create their own. • The algorithms pupils create can often be implemented using programmable toys or "human robots", and it can be useful for pupils to compare how a square is drawn with a floor turtle and with Logo or ScratchJr. • As the children break down larger tasks into smaller instructions, they also develop their use of decomposition to solve a problem.	• Using storyboards or flowcharts, simple rough jottings or "pseudocode" (a written description of how a program will operate). • pupils apply algorithms as rules to their understanding of maths and grammar. They might create algorithms for how to multiply by 10, 100 and 1,000. They increase their understanding of algorithms as sequences of instructions – perhaps writing one for how to play their favourite sport – and they build their understanding of how algorithms and programs are distinct but intrinsically linked. • Pupils are expected to think algorithmically, using decomposition confidently, to design programs with particular goals. They should be able to debug them.
Examples in other areas of the curriculum	Instructional writing in English, the method for a science experiment: each can be considered an algorithm. PE – getting dressed for a PE lesson (following a sequence of steps) Maths – children's approach to mental arithmetic might be an implementation of a simple algorithm.		
Decomposition 	• When children label simple diagrams and sequence familiar processes, they start to see that breaking things down into their parts helps us learn about them. • In roleplay, children could think about how to set up a shop – they'll need merchandise, price tags, a till, money for change, etc. • Constructing a model plane, they make the wings, add these to the body, add the wheels: the children think about the parts and then assemble them. • It is important to model these skills and take them a step further by showing how to evaluate that all the necessary things are present.	• Model how to take this further by encouraging children to evaluate whether they've missed aspects and to share their understanding with others.	• Decomposition occurs in planning and collaborating. Provide opportunities for pupils to collaborate as a team. • Pupils need to take a topic and decompose it into its various aspects and then work collaboratively to develop a wiki page for exam[ple or presentation • Pupils decompose projects into phases of planning, research, drafting, reviewing and publishing.

Examples in other areas of the curriculum	Any task or project will need to be decomposed into smaller, more-manageable parts. Decomposition is everywhere. Humanities - concept maps are more detailed. In exploring detail, children increase their awareness and independence. Science - children should have ongoing opportunities to break things down into their constituents – e.g. a lifecycle and its stages, a diagram and its labelled parts.		
Patterns 	• Children are given practical situations where they can notice patterns, observing and exploring similarities and differences. • They can be presented with sets of items which are sortable in various ways. • For example, they could be given a water tray and assorted objects, some of which float.	• Children continue to engage in practical experiences where similarities and differences can be explored. The range and complexity of these scenarios increase. • Model how to notice patterns, how to think of rules and how to try them out.	• Pupils can identify patterns and rules in number sequences; and use logical reasoning to explain • Eg In maths and science, they might look for similarities and differences in numerical data to answer questions, draw conclusions and make predictions. • Pupils will explore theories using predict-and-test activities eg in geography, pupils might be asked what constitutes a good location for a new town. To formulate ideas about town planning, they could look at patterns of existing settlements and test those ideas on maps. • Pupils can spot repetition of code
Examples in other areas of the curriculum	Children become familiar with repeated phrases in nursery rhymes. Reading - children notice repeated structures in stories. Music - repeating lines in many musical forms Maths - children typically undertake investigations in which they spot patterns and deduce generalised results, look for patterns in more-abstract concepts, including odd and even numbers, negative numbers, multiples and inverse operations. English - children spot more-complex spelling patterns, and they listen for patterns in sounds (phonemes). Science – group and classify – children will notice rules and patterns, for example in animals' appearance and habitat or in the properties of materials, and they'll draw on those patterns to make predictions in other investigations.		
Abstraction 	• Opportunities to summarise as children remember events and recount what was important. • In maths, they start to sense the abstraction of number: they can count three bears, three bricks, three friends, etc and formulate an abstraction of 'three-ness'.	• Starting to identify the important elements and ignoring unnecessary detail.	• Pupils continue to become more experienced in abstraction. • Pupils can reflect on what they know and create summaries in pre- and post-topic assessments for example, recording the most-important facts and so creating an abstraction of their understanding. • Pupils can compare and contrast • Pupils will focus on the representation of key information whilst leaving to one side many details
Examples in other areas of the curriculum	Maths - working with word problems often involves identifying key information and thinking how to represent it in the more abstract language of arithmetic. Music - abstracted to notation. Geography - learning how to add places of interest and to ignore detail; they use world maps and create local maps and so start to see different layers of abstraction. Children can also gain experience of abstraction when playing computer games, appreciating that these interactive simulations are based on real life, but are simpler. History - children consider viewpoints as they roleplay famous people		
Evaluation 	• Children can start to develop their evaluation skills as they articulate their judgements and reasons in simple terms, such as "My dog is my favourite pet because she lets me pat her." • Children to consider different ways to find out things. They can find out about dinosaurs by reading a text, browsing a picture book, using a CD ROM or entering keywords into a search engine.	• Children express preferences more readily and clearly • Children can undertake many different computing activities which include simple evaluation. They can be introduced to the idea of design goals and criteria and may begin to create their own. • Designing **algorithms** for a Bee-Bot moving between two points, they can evaluate the most effective route, for example the shortest. • They can have criteria for designing a Bee-Bot maze – a start, an end, a minimum number of obstacles – and for the Bee-Bot itself: they might want it to navigate the maze without striking anything. • Children can refer to the design criteria and judge if they've been met.	• Pupils begin to understand the importance of evaluation in improving work, and they take more responsibility for it. • With increasing confidence and independence, they use more-detailed design goals and criteria and become more comfortable in drawing up their own. • They become more skilled in giving and absorbing appropriate feedback.
Examples in other areas of the curriculum	Self- and peer-assessment can help to develop children's evaluation skills, as they make judgements using success criteria and consider potential improvements. PE - a list of 'good' things to aspire to – perhaps certain moves in a routine, perhaps landing on two feet. English - the success criteria for a child's written work might be the correct use of capitals and full stops, or the inclusion of adjectives and adverbs. They may recommend a book to a friend, explaining why they think it will be enjoyed, having made a judgement about what type of books might be favoured. Design Technology - makes use of evaluation as pupils work through the design–make–evaluate cycle.		

SIGNIFICANT PEOPLE

YEAR 1	YEAR 2	Year 3	Year 4	Year 5	Year 6
Ada Lovelace She is credited with being 'the first programmer'. She worked with Charles Babbage to create the first alogorithm intended to be processed by a machine. **Sergey Brin and Larry Page** Invented GOOGLE search engine whilst at university.	**Margaret Hamilton** Margaret Hamilton is a computer scientist. She designed software for Apollo 11 (the spacecraft used in the 1969 Moon landing). Her software helped to prevent the Moon landing from being cancelled! **Jerry Lawson** Electronic engineer that lead the team at Fairchild that pioneered the commercial video game cartridge. Dubbed by some as the "Father of Modern Gaming."	**Charles Babbage** He was an English mathematician, analytical philosopher, mechanical engineer and computer scientist. He was the first person to invent the idea of a computer that could be programmed. **Annie Easley** One of the first African-Americans computer scientists at NASA. Leading member of the team that developed the software for the Centaur rocket stage.	**Tim Berners-Lee** He is widely credited with inventing the internet we know and use today. The World Wide Web (WWW) gives users access to unlimited communication and documentation without relying solely on email. **Mary Jackson**. She was the first African-American female engineer at NASA, Jackson dealt with the common-day segregation and became one of the 'Hidden Figures'	**Guido Van Rossum** In the early 90s, Van Rossum developed Python, one of the most popular coding languages used by programmers & by companies like Spotify & Dropbox. Python powers a huge variety of software programs used today. **Grace Hopper** She invented the first computer programming language to use English and the first computer compiler which translated written instructions into codes that computers could read.	**Barbara Liskov** Throughout her career, Barbara Liskov focused on programming methodology, developing and implementing CLU and Argus languages, with a specialty in programming systems and distributed computing. **Alan Turing** He made major contributions to the fields of mathematics, computer science, and artificial intelligence. He worked for the British government during World War II, when he succeeded in breaking the secret code

				She told people that one day computers would be small enough to fit on a desk. At the time, they took up whole rooms!	Germany used to communicate. He suffered from homophobic discrimination.