Cabot Learning Federation

I am a computer scientist

I am a computer scientist. I study computers and computing, including their theoretical and algorithmic foundations, hardware and software, and their uses for processing information. As a computer scientist, I am developing innovative ways of exploring information, and what I am able to do with it: enhancing my own thinking skills through algorithms, logic, visualisation, precision, and abstraction. I understand that computational thinking involves and illuminates other disciplines and the collaboration of these are vital, when addressing the advances in technology, and the inventive ways to use it. I am interested in pushing the boundaries of invention and transformation, using technology to creatively solve problems around me.

	Key Concepts	Contexts
What is computing? Algorithms have been used for around 4000 years, with evidence in Sumerian tablets displaying examples of division. People have used tools , such as an abacus, to help with logical thinking for 2000 years. Over time, these tools and algorithms have developed in complexity and the first modern computer was designed by Charles Babbage in the 1820s and 30s. Slow and massive machines have developed into more efficient and compact devices we know today. The versatile nature of this technology means that it is in constant use and is ever changing but in essence runs on similar mathematical principals as the original machines. The study and use of these machines and their algorithms is computer science.	**Computer components and resultant hardware** Computers are electronic devices used for storing or processing data (information). They can be used by inputting, storing, and outputting data. To do this, computer programs are written by coders (programmers) and can be very sophisticated. Therefore, computers do not think and cannot make decisions by themselves, rather the software (programs to do a specific job) enables users to interact to give (input) and receive (output) data.	Y1: Using computers Y2: How computers work and the internet Y3: Global computers Y4: Networks and data Y5: Computing & Security Y6: Compatibility and design
	Computational Thinking • Understanding and using algorithms: a sequence of specific, ordered instructions to complete a task. • De-bugging: finding and resolving bugs (defects or problems that prevent the correct operation • Abstraction: removing or filtering out all unnecessary information to focus on what we need to solve a problem • Decomposition: breaking down a problem into smaller, easier to solve parts • Pattern Recognition: recognising patterns within outcomes.	Y1: Introduction to algorithms Y2: Knowing and exploring algorithms Y3: Programming Y4: Being a Programme Engineer Y5: Complex Programming with conditions and patterns Y6: Mastering Programming
	Digital Literacy relates to having functional IT skills, which support the creation or presentation of content. To be digitally literate is to have access to a broad range of practices and cultural resources that you are able to apply to digital tools. It is the ability to make and share meaning in different modes and formats; to create, collaborate and communicate effectively and to understand how and when digital technologies can best be used to support these processes.	Y1: Word processing; Working with sound, images and video Y2: Word processing; Animation Y3: Working with sound, images and video; Handling Data Y4: Animation Y5: Presenting information online; Handling Data Y6: Working with sound, images and video
	Technology should be used safely, respectfully, and responsibly. Electronic safety (e-safety) incorporates: • Safe, respectful, responsible use of technology • Identifying dangers • Know where or who to contact if they have concerns	E-safety is explicitly taught within the CLF Citizenship/PSHE curriculum. Opportunities to develop pupils e-safety are expected to be referred to within the teaching of Computer Scientists.

Key Concepts, Knowledge, Vocabulary and Skills – Computer scientists: Year 1

Key Concept	Key Knowledge and Vocabulary	Skills: Techniques & Application
Computer components and resultant hardware	**Using computers** • Identify technology e.g. traffic lights, pass entry systems, • Identify a computer and its main parts e.g. monitor/screen, keyboard, central processing unit, mouse/navigator, speakers • Computers can do nothing, unless instructed by a user • Data is collected and can be stored on a computer • A network is two or more computers that are linked	**Using computers** • Turning devices on/off safely • Using the equipment in a safe and respectful manner • Troubleshoot the computer/device for errors e.g. checklist for start up: monitor on, computer on
Computational Thinking: • **Understanding and using algorithms** • **De-bugging** • **Abstraction** • **Decomposition** • **Pattern Recognition**	**Introduction to algorithms** Algorithms: • An algorithm is a series of set of instructions • Know the sequence of an algorithm is important De-bugging • Debugging is identifying and removing errors from an algorithm. Abstraction: • Abstraction is taking out instructions you do not need in an algorithm Decomposition: • Decomposition is solving a problem in smaller parts to make it easier Pattern & Recognition: • Know which parts of instructions/algorithms are similar and different	**Introduction to algorithms** Algorithms: • Follow simple sequences of instructions • Follow instructions given by someone else • Explain what a given command, within an algorithm, will do • Write simple algorithms e.g. arrow directions, mathematical calculation 6 + 4 =, symbols for real life experiences De-bugging • Debug simple algorithms Abstraction: • Remove unnecessary instructions you do not in an algorithm Decomposition: • Identify different parts of a problem e.g.. *brushing teeth (getting equipment, brushing, cleaning up)* Pattern & Recognition: • Predict the behaviour of simple programs, with or without errors • Describe the similarities and differences in instructions/algorithms
Digital Literacy relates to having functional IT skills, which support the creation or presentation of content.	**Word processing** • See computer components key knowledge and vocabulary • There are words commonly used in in word processing: ○ Undo – To undo the last known command or action that the program was given ○ Format – To change the presentation of the information, e.g. layout or text ○ Edit – To make specific changes, including corrections to the presentation ○ Cursor – The cursor is an indicator of the current position for user interaction e.g. arrow ○ Font – The style of the words in the presentation	**Word processing** • Type using a keyboard, locating letters, numbers, keys, and some punctuation (?!.), recognising two hands are required. • Explore different key features on a keyboard e.g., the space bar to make space, delete to delete letters and/or words / enter to make a new line and/or return • Use a shift key for punctuation including capital letters • Make the arrow keys move the cursor • Use 'undo' to remove changes • Move cursor and left click to select, using a mouse and trackpad, including selecting some / all the text by clicking and dragging • Use text within work • Edit and improve my work by changing, adding, or removing words • Use tools to edit - font size, colour, and style
	Working with sound, images and video: • Devices can capture photos of real life • Devices can capture a video of real life • Sounds can be recorded on digital devices	**Working with sound, images and video:** • Take a photograph using a digital camera • Record a video using a camera application • Record sounds
Technology should be used safely, respectfully, and responsibly.	Electronic safety (e-safety) incorporates: • Safe, respectful, responsible use of technology • Identifying dangers • Know where or who to contact if they have concerns E-safety is explicitly taught within the CLF Citizenship/PSHE curriculum. Opportunities to develop pupils e-safety are expected to be referred to within the teaching of Computer Scientists.	

Key Concepts, Knowledge, Vocabulary and Skills – Computer scientists: Year 2

Key Concept	Key Knowledge and Vocabulary	Skills: Techniques & Application
Computer components and resultant hardware	**How computers work and the internet** • Know that computers have no intelligence and that computers can do nothing unless a program is run. • Know the purpose of a network and why they are computers are networked. • Recognise different storage tools, e.g. memory sticks, external hard drives, cable connections i.e. USB. • The internet is a global network of computers, including online devices and servers • The World Wide Web is the collection of public webpages stored on the internet • Some websites are blocked or have limited access because of network administration permissions e.g. parental and school controls • Explain the roles within a computer system for hardware and application software: - Computer memory and processing speed - Size of files and storage available - There are operating systems and these may look differently - Understand where files are stored and saved • Be aware that websites and videos are files that are shared from one digital device to another. • Know that computers collect data from various input devices, including sensors and application software., e.g. digital camera, mouse roller, gate/motion sensors, temperature and weather sensors	**How computers work and the internet** • Troubleshoot the computer/device to identify possible errors such as connections to networks, e.g. symbols for wi-fi, online/offline, battery power • Seek technical help for an error message e.g. "cannot connect to the server". • Correctly insert and remove different storage tools to ensure data in not corrupted. • Maintain a safe and organised working environment e.g. keep liquids and consumables away; cables to be managed for safety.
Computational Thinking • **Understanding and using algorithms** • **De-bugging** • **Abstraction** • **Decomposition** • **Pattern Recognition**	**Knowing and exploring algorithms** Algorithms: • An algorithm is a series of set of instructions • Know the sequence of an algorithm is important • Be aware algorithms can be represented in simple formats [flow chart, storyboards, and narrative text] De-bugging • Debugging is identifying and removing errors from an algorithm. Abstraction: • Abstraction is taking out instructions you do not need in an algorithm Decomposition: • Decomposition is solving a problem in smaller parts to make it easier Pattern & Recognition: • Know which parts of instructions/algorithms are similar and different Note: The Year 2 key knowledge and vocabulary is the same as Year 1	**Knowing and exploring algorithms** Algorithms: • Write algorithms for everyday tasks e.g. • Give clear and unambiguous instructions, choosing a series of words that can be enacted as a sequence • Use an algorithm to program a sequence • Express algorithms using icons • Incorporate loops, arithmetic operator and *if statements* when creating algorithms with support • Include loops to make code more efficient • Write algorithms e.g. range of icons and loops, 5+5+5 = or 3X5 = , journey to school • Create a design for a program which includes backgrounds and characters • Explore outcomes by suggesting different orders and discussing the outcomes of the algorithm De-bugging • Debug programs of increasing complexity e.g. use icons, incorporate loops, arithmetic operator and if statements • Explain what happens when we change the order of instructions Abstraction: • Identify the important information within a sequence of events or program. • Identify errors or ways to streamline a program through debugging Decomposition: • Explain decomposition is the breaking objects and/or processes down • Use logical thinking to explore software, predicting, testing, and explaining what it does e.g. Pattern & Recognition: • Use logical reasoning to predict the outcomes of algorithms • Identify similarities and differences between algorithms and the outcomes • Show the difference in outcomes between two sequence that consist of the same commands • Correct algorithms to match intended outcomes
Digital Literacy relates to having functional IT skills, which support the creation or presentation of content.	**Word processing** • Know what is meant by the home, bottom and top rows when typing. • Know the shift and caps locks in capital letters • Copy and paste means to select and copy text or images and relocate them within or to another document • Know the indicators of programme errors in word processing e.g. lines beneath words for spelling errors • Know shortcuts are commands to assist speed in function skills • Know the universal symbols within operating platforms e.g., cursor arrow, saved disk, processing/thinking	**Word processing** • Type with awareness of home, bottom and top rows • Use the shift and caps locks in capital letters • Use the space bar only once between words and use touch to navigate to words / letters to edit • Move the cursor using arrow keys and insert text and/or make changes • Using 'back space' and delete interchangeably • Click and drag to move items / highlight text and use arrow keys to edit • Copy and paste images and text • Use shortcuts for undo, redo, copy and paste • Use spell check features to check text • Incorporate different tools: apply font style, minimise, and maximise pages, page set up and align text
	Animation • Animation is sequence of still scenes/images that can create a sense of movement when viewed in rapid succession • Stop-film animation is a term to describe capturing a photograph	**Animation** • Explain how an animation / flip chart works • Create an animation to tell a story with more than one scene • Create multiple animations of an image and edit these together • Design and produce a simple stop-film motion animation
Technology should be used safely, respectfully, and responsibly.	Electronic safety (e-safety) incorporates: • Safe, respectful, responsible use of technology • Identifying dangers • Know where or who to contact if they have concerns E-safety is explicitly taught within the CLF Citizenship/PSHE curriculum. Opportunities to develop pupils e-safety are expected to be referred to within the teaching of Computer Scientists.	

Key Concepts, Knowledge, Vocabulary and Skills – Computer scientists: Year 3

Key Concept	Key Knowledge and Vocabulary	Skills: Techniques & Application
Computer components and resultant hardware	**Global computers** • Recognise the components within digital devices [processor, memory storage, motherboard, display, speakers, microphone – other input devices webcam, finger-print scanner] • Recognise what computers do when they receive instructions that can not be executed [error messages] • Recognise one or more computing language [binary, html (hyper-text markup language) • Understand how computers can be connected to form a network either wirelessly or via LAN • The internet is a global network of computers, including online devices and servers • The World Wide Web is the collection of public webpages stored on the internet [understand URLs and domains). • URL means Universal Resource Locater and is an address to locate data • A domain is the name of a website e.g. google or BBC bitesize • Understand that copyright exists and that they are whole libraries of content available for public use	**Global computers** • Troubleshoot the computer/device to identify possible visual errors such as connections to devices, e.g. microphone, speakers, camera, printers • Locate general device information e.g., wifi network, storage capacity and usage, which programmes are installed etc • Recognise when a problem can be solved individually without 'technical support'. • Use the language URL and domain to describe the location of information and identify if there are errors with the domain or url • Relate domains and urls to wider world context e.g. home address, room locations in a building
Computational Thinking • **Understanding and using algorithms** • **De-bugging** • **Abstraction** • **Decomposition** • **Pattern Recognition**	**Programming** Algorithms: • Different algorithms work for different purposes; there are patterns • Programs are executed by following precise and unambiguous instructions • It is possible to predict, and test completed algorithms De-bugging • Debugging is necessary for all algorithms for a variety of situations • A systematic approach to debugging means describing, explaining and correcting errors Abstraction: • A sequence of events will have more important information • Information is not always needed for the sequence of events and so some instructions can be removed. Decomposition: • Decomposition can be explored without computers (unplugged activity): Visual representation, including mathematics can be used to solve problems. Pattern & Recognition: • Patterns can be recognised and included within a program	**Programming** Algorithms: • Identify objects in a project • Program movement • Recognise commands can be represented in different ways, including as blocks • Create a program following a design • Start programs in different ways • Combine sound commands • Know objects can be controlled by the commands I choose De-bugging • Review programs and algorithms to identify improvements and locate errors • Identify a way to improve a program Abstraction: • Choose which keys to use for actions • Discuss the requirements of an algorithm / program Decomposition: • Create a sequence of connected commands • Implement my algorithm as code Pattern & Recognition: • Experiment with input peripherals • Explain what a sequence is and use this in programmes • Identify additional features including patterns and repetition
Digital Literacy relates to having functional IT skills, which support the creation or presentation of content.	**Handling data** • Data is facts and information collected for a purpose usually expressed as numbers, usually used with a computer • Data is collected for a variety of purposes, and this can be collected responsibly or irresponsibly i.e. with permission or not • Data can be considered more important than others, e.g. personal information • Spreadsheets is table to organise information. This is usually used for numerical information. • A cell is single point in a spreadsheet that holds one piece of information. This is usually a number. • A formulae is an instruction. It is like an algorithm. • There are formulas for functions: sum, count • Sorting means to organise the data by an order e.g. alphabetically, largest to smallest.	**Handling data** How data can be used: • Identify real life scenarios where data is collected and interpretated e.g. weather Collecting data: • Identify a line of enquiry that requires data in order to answer it e.g. "The older you are the taller you ." • Review and select the appropriate method of collecting data • Collect data with paper/pencil and transfer to a computer program • Navigate the data handling software e.g. excel • Add text and numbers to spreadsheet cells with accuracy • Use formatting skills to aid organisation and presentation e.g. adding borders, bolding titles, colouring cells. Sharing data: • Discuss the data that has been shared, linking to the line of enquiry. Analysing data: • Use functions to explore and highlight important data e.g. selecting sort or auto sum icons/tools • Answer questions relating to data collected and pose my own questions. • Notice patterns and trends in the data e.g. similarities and differences, most, least
	Working with sound, images, and video: • Devices (which capture photo, video, and sound) can be used to create a mixed media project • Photographs, videos, and sounds can be edited (using different tools) to create new digital artwork • A frame in a video is equivalent to a static / still image • Cropping means removing sections of video or images • Visual effects are the creation or adjustment to imagery, including video • Transitions are any visual changes between two frames/scenes - e.g. cuts, dissolves, fades and wipes.	**Working with sound, images, and video:** • Create a mixed media outcome which contains audio, video, and music • Record a film with a clear brief, reviewing the creation noting areas for improvement, and reorder the film with the necessarily changes. • Manipulate and or adjust images with effects: colour, features, framing etc
Technology should be used safely, respectfully, and responsibly.	Electronic safety (e-safety) incorporates: • Safe, respectful, responsible use of technology • Identifying dangers • Know where or who to contact if they have concerns E-safety is explicitly taught within the CLF Citizenship/PSHE curriculum. Opportunities to develop pupils e-safety are expected to be referred to within the teaching of Computer Scientists.	

Key Concepts, Knowledge, Vocabulary and Skills – Computer scientists: Year 4

Key Concept	Key Knowledge and Vocabulary	Skills: Techniques & Application
Computer components and resultant hardware	**Networks and data** • There are a range of connected networks e.g. home network, school, organisation, national and intranets • Computer networks provide multiple services, such as the world wide web and opportunities for communication and collaboration. • There are types of different network hardware where computers are linked e.g. desktop computers, mobile devices, servers, routers, • A server is a computer or system that stores all the data of a website to allow access from another device (clients) and its user e.gl. cloud storage. • An IP address (internet protocol address) is an unique address that identifies a device on the internet or a local network, using numbers. • Digital computers use binary to represent all data. • Packets of data are units of **data** made into a single package that travels along a given network path.	**Networks and data** • Identify the range of connected networks e.g. home network, school, organisation, national and intranets • Describe and design/map(draw) real and fictional networked devices and how they connect through websites, web pages, apps and IPs • Locate information about storage capacity on devices e.g. file Explorer. • Store and share data and understand the role of packets • Identify suitable storage locations for own information: e.g. saving files onto personal or school networks
Computational Thinking • **Understanding and using algorithms** • **De-bugging** • **Abstraction** • **Decomposition** • **Pattern Recognition**	**Being a Programme Engineer** A programme engineer will design using programming language for an intended purpose. Algorithms: • Understand that a program is a sequence of statements written in programming language. • Algorithms for use when programming can become increasingly more precise. • Remix existing code De-bugging • Being systematic and applying logical reasoning can assist in the detection and correction of errors in programs Abstraction: • Abstraction (removing of instructions) can help to focus on what is important in a design • Abstraction (removing of instructions) ad pattern recognition can help to modify code Decomposition: • Decomposition can be explored without computers (unplugged activity): Visual representation, including mathematics can be used to solve problems Pattern & Recognition: • Procedures, pattern matching, and simple selection can be used in algorithms • Know that a computer can repeatedly call a procedure A procedure is set of coded instructions within the program • Patterns are similarities or characteristics that some of the problems share • Pattern recognition involves finding the similarities or patterns among small, decomposed problems that can help us solve more complex problems more efficiently • Simple selection is a decision or question. In an algorithm, there may need to be a question because the algorithm has reached a step where one or more options are available. Depending on the answer given, the algorithm will follow certain steps and ignore others	**Being a Programme Engineer** Algorithms: • Program a computer by typing commands • Explain the effect of changing a value of a command • Create a code for a given purpose • Write an algorithm to produce a given outcome • Test my algorithm in a text-based language • Make use of a design to write a program • Predict the outcome of a snippet of code • Recognise some programming languages enable more than one process to be run at once De-bugging • Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems Abstraction: • Modify a snippet of code to create a given outcome Decomposition: • Change loops to produce a given outcome • Identify which parts of a loop can be changed Pattern & Recognition: • Identify repetition and patterns • Use a count-controlled loop to produce a given outcome • Identify the effect of changing the number of times a task is repeated • Choose when to use a count controlled and an infinite loop
Digital Literacy relates to having functional IT skills, which support the creation or presentation of content.	**Animation** • Animation is a sequence of still images that can be planned, scripted, manipulated and edited • Different animation types: tradition / hand drawn animation; stop-film; 2D, 3D, motion graphics	**Animation** • Use software to plan, script and create 3D animated stories • Improve and develop stop motion animation clips with techniques such as onion skinning • Editing sections of animation videos together to create a continuous piece • Export animations as GIFs or videos
Technology should be used safely, respectfully, and responsibly.	Electronic safety (e-safety) incorporates: • Safe, respectful, responsible use of technology • Identifying dangers • Know where or who to contact if they have concerns E-safety is explicitly taught within the CLF Citizenship/PSHE curriculum. Opportunities to develop pupils e-safety are expected to be referred to within the teaching of Computer Scientists.	

Key Concepts, Knowledge, Vocabulary and Skills – Computer scientists: Year 5

Key Concept	Key Knowledge and Vocabulary	Skills: Techniques & Application
Computer components and resultant hardware	**Computing & Security** • There are a range of computing technology that is used as part everyday life e.g. contactless payments, mobile apps, entry systems, security surveillance systems • The security of data relates to passwords, access privileges/rights and data encryption • Data encryption means the process of converting data into an unusable form (but knowing this does not itself stop hacking or data theft). • Malware (short for malicious software) means software which is designed to harm or exploit any programmable device, service or network (e.g. viruses, ransomware and spyware. • A computer virus is a type of malicious code or program written to alter the way a computer operates and that is designed to spread from one computer to another • Antivirus is a kind of software used to prevent, scan, detect and delete viruses from a computer.	**Computing & Security** • Check network connection and availability for owned devices, including 'levels of access' and the source of connection • Identify network speed and how to improve connection • Explain the impact a server may have on a connection speed and suggest improvements. • Transfer data safely and securely across the internet with an understanding of data protection rules, including encryption of messages • Describe and map(draw) real and fictional networked devices and the intended security and access privileges/rights, including home and organisational access, as well as software program installation rights. • Identify unsafe networks or websites, including signs of malware or viruses
Computational Thinking • **Understanding and using algorithms** • **De-bugging** • **Abstraction** • **Decomposition** • **Pattern Recognition**	Complex Programming with conditions and patterns Algorithms: • Write complex algorithms for a purpose, evaluate my work, and identify errors. • Include a sequence, selection, and repetition commands, combined with variables as required, to implement a design. • The flow of a program means… • Conditions are a state or situation something is experiencing • Conditionals Statements - A way for computers to make decisions based on conditions. If/Else are common forms of conditional statements in programming; *tells the computer that if the condition is true, do this. Else, if the condition is false, do another thing.* • Understand that websites can be altered by exploring the code beneath the site De-bugging • Being systematic and applying logical reasoning can assist in the detection and correction of errors in programs. Abstraction: • Understanding abstraction (removing of instructions) can help to identify errors in code i.e. what has been removed accidentally or deliberatively. Decomposition: • Understand decomposition is and how it facilitates problem solving of increasing complex problems. e.g. Pattern & Recognition: • Include patterns and selection to create specific outputs in algorithms. • Compare various programmes to identify commonalities / similarities. • A count-controlled loop repeats a series of one or more instructions until a determined number of repetitions of the loop/sequence has been completed. • An infinite loop is a sequence of instructions in a computer program which loops endlessly, either due to the loop having no terminating condition • A conditional loop is a sequence of instructions in a computer program which repeats set of commands depending on the conditions set.	**Complex Programming with conditions and patterns** Algorithms: • Use logical reasoning to explain how simple algorithms work • Create a simple circuit and connect it to a microcontroller • Program a microcontroller • Connect more than one output component to a microcontroller • Identify a condition and an action in a project • Create a program with different outcomes using selection and describe what it will do • Explain that program flow can branch according to a condition • Design the flow of a program • Identify the setup code I need in my program De-bugging • Test and debug a project • Identify ways the program could be improved • Interpret other's existing code to explore a problem, suggesting ways for improvements. Abstraction: • Acting out and analysing plans can help to eliminate unnecessary actions. • Review instructions to see if abstraction is required or has taken place. Decomposition: • Use decomposition to explore code • Solve problems by decomposing them into smaller parts • Identify and modify conditions in a program • Change code to achieve a personalised outcome Pattern & Recognition: • Include patterns and selection to create specific outputs in algorithms. • Use a count-controlled loop to control outputs • Design sequences that use count-controlled, conditional, and infinite loops
Digital Literacy relates to having functional IT skills, which support the creation or presentation of content.	**Handling data** • Collect data and identify where it could be inaccurate • Discrete data is data that can only take specific values e.g. shoe size • Continuous data is data that can take any value or range of value e.g. height • There are different graphs that can be used for different purposes. These include bar charts, pie charts, pictograms, line graphs, scatter graphs • Formulae: count, sum, average, sum conversion, countif, • Filtering means to show or hide particular data e.g. *numbers greater than 25* • Conditional formatting is a visual form of filtering and highlights particular data e.g. *show cell in red if greater than 2* • Interpretating means drawing conclusions. • A database is a collection of structured information e.g. a database on land based animals • Databases hold information in fields. A field is single point to hold one piece of information used in a database. This is usually in words.	**Handling data** How data can be used • Identify real life scenarios where data is collected and interpretated, which is known or unknow to them e.g. pupil voice, number plate car tracking Collecting data • Identify a line of enquiry that requires data in order to answer it • Review and select the appropriate method of collecting data and how this could be presented. Knowing how the graph 'pulls data' will help determine the arrangement of data collection. • Identify the type of data that is being collected e.g. discrete and continuous and know the permitted range for data entry e.g. enter a number between 10 and 20. • Add text and numbers to spreadsheet cells • Use formatting skills to aid organisation and presentation e.g. number formatting (£, decimals, date). data validation • Organising data in a systematic way entering common characteristics of the data, including labelling columns and rows e.g. numbers for height measurements Sharing data • Collect data using an electronic device and appropriate program e.g. excel • Choosing suitable graphs and the associated detail that is shared for a particular audience • Present data in a graph, selecting the most appropriate layout. • Selecting the most suitable representation for the data e.g. line graph to represent continuous data • Presenting data to an audience and sharing the answer to a chosen enquiry Analysing data: • Answer questions relating to graphs and pose my own questions. • Extract and retrieve data from a range of charts: range, outlying data • Where appropriate, calculate range, average (mean) • Interpretating data identifying patterns, correlations • Use skills to scrutinise the data: sorting, filtering, conditional formatting • Use formulae to further investigate data (count, sum, average, sum conversion, countif) e.g. count… *which values are greater than 10% from the average*

	Presenting information online • An ebook is short for "electronic book" and uses either a computer, mobile device or ebook reader to display the content. • Webpages are created using code known as hmtl • Html is an acronym for HyperText Markup Language (HTML). HTML is the set of codes inserted into a file intended for display on the Internet. • Embedding means to place content, usually from a different website, onto a webpage rather sharing a webpage linking.	**Presenting information online** • Create an eBook with text, images, and sound. • Design and publish a webpage and embed videos. • Make and export an interactive presentation including a variety of media, animations, transitions, and other effects. • Create an interactive guide to an image by embedding digital content and publishing it online. • Understand purpose e.g. inform, entertain or persuade • Produce a website / web app which includes a variety of media • Be able to reflect on the effectiveness of a presentation and seek feedback
Technology should be used safely, respectfully, and responsibly.	Electronic safety (e-safety) incorporates: • Safe, respectful, responsible use of technology • Identifying dangers • Know where or who to contact if they have concerns E-safety is explicitly taught within the CLF Citizenship/PSHE curriculum. Opportunities to develop pupils e-safety are expected to be referred to within the teaching of Computer Scientists.	

Key Concepts, Knowledge, Vocabulary and Skills – Computer scientists: Year 6

Key Concept	Key Knowledge and Vocabulary	Skills: Techniques & Application
Computer components and resultant hardware	**Compatibility and design** • There are links between internal parts within a computer, to ensure compatibility. • Compatibility is… • Not all hardware/components are compatible with each other (Window, Apple, *Linux,* Chrome) app compatibility • Computer software develops solve problems of compatibility, *i.e. compatible with Windows* • Updates refer to software improvements that are made and support the functionality of the software and hardware effectiveness **Servers** • A private server is any computer used as a server that is privately administrated. • A public server is data can be accessed by others who are also connected to it. • Data transmission between digital computers over networks, including the internet i.e., IP addresses and packet switching. • An IP address is a unique address that identifies a device on the internet or a local network. IP stands for "Internet Protocol," which is the set of rules governing the format of data sent via the internet or local network. • Packet switching is the transfer of small pieces of data across various networks. These data chunks or "packets" allow for faster, more efficient data transfer, i.e W*hen a user sends a file across a network, it gets transferred in smaller data packets, not in one pieces.*	**Compatibility and design** • Link different parts of computer hardware to achieve an outcome (eg: raspberry pi, maKey-maKey) • Recognise where hardware components are compatible and not compatible. • Critically analyse different networks, viewing their purpose and suitability for specific outcomes. • Describe and design(draw) real and fictional networks, viewing their purpose and suitability for specific outcomes. • Recognise the need, rights/permissions and need to run software updates, which in turn support hardware
Computational Thinking • **Understanding and using algorithms** • **De-bugging** • **Abstraction** • **Decomposition** • **Pattern Recognition**	**Mastering Programming** Algorithms: • Increasingly complex algorithms are written for purpose • Generic code can be written for use across multiples projects. • A variable are storage locations with assigned names (categories) that hold data that you can change or use later. • Controllable device are ones which can be programmed and controlled. De-bugging • Editing code contributes to unsuccessful and successful outputs and applications • A software bug is an error in a programme that causes it to produce an incorrect or unexpected result. The elimination of software errors is called bug fixing. • A bug fix is a change to a system or product designed to handle a programming bug/glitch. Abstraction: • Logical reasoning can help to explain how processes can be more efficient, reducing the number of potential steps needed in a program. • Knowing abstraction is part of the planning process before creating algorithms. Decomposition: • Use decomposition across a range of situations, to break down complex programs, making improvements where required. Pattern & Recognition: • Recognise, and make use, of patterns across programming projects, identifying variables needed, and explain their uses in selection and repetition. • A nested loop is a loop within a loop, an inner loop within the body of an outer one • Iteration is the repetition of a process such as a loop.	**Mastering Programming** Algorithms: • Make use of an event in a program to set a variable • Recognise that the value of a variable can be used by a program • Transfer a program to a controllable device • Use a variety of software to write code • Understand that websites can be altered by exploring code beneath the site • Write code to create a desired effect, using python. • Evaluate increasingly complex code. De-bugging • Use a range of approaches to find and fix bugs • Modify a program to achieve a different outcome • Design, write and debug programs that accomplish specific goals • Demonstrate skills by debugging problems with different inputs and outputs • Interpret other's existing code to explore a problem, editing to ensure successful application. Abstraction: • Simplify programs and algorithms. Decomposition: • Use a variable if an if, then, else statement to select the flow of a program • Determine the flow of a program using selection • Use a condition to change a variable • Solve problems by decomposing them into smaller parts Pattern & Recognition: • Identify examples of information that is variable • Use sequence, selection and repetition in programs • Use loops, including nested loops.
Digital Literacy relates to having functional IT skills, which support the creation or presentation of content.	**Working with sound, images and video:** • Green screen is a technique used to create special effects through the use of a green backdrop (large or small scale). • A WEBMs is an audio-visual media file format which is designed for sharing files. • GIF is a graphic interchange format which presents as a short, looped animation. • Editing is used to remove unwanted elements, join different parts, adjust the geometry or to enhance the effects of the image. video or sound. • Photo editing is the process in digital photography of making adjustments to photographs in a photo editing program	**Working with sound, images and video:** • Create a video of high quality, using a range of media – green screen, animations, film, and image • Incorporate different tools in video creating including subtitles, crop, resize • Create a purposeful multi-layered digital image, for a specific design brief • Evaluate and discuss images explaining effects and filters that been used to enhance the media • Create WEBMs and MP4 with sound GIFS • Create different sound GIFs • Store, retrieve and export media to a computer • Explore photo editing capabilities, detailing changes made to an image
Technology should be used safely, respectfully, and responsibly.	Electronic safety (e-safety) incorporates: • Safe, respectful, responsible use of technology • Identifying dangers • Know where or who to contact if they have concerns E-safety is explicitly taught within the CLF Citizenship/PSHE curriculum. Opportunities to develop pupils e-safety are expected to be referred to within the teaching of Computer Scientists.	

Appendix 1: Curriculum Rationale

Why have particular contexts been chosen? Why is it organised in this way? Why will it help children?
The answers to these questions are rooted in the rationale in the design of the curriculum.

This curriculum is coherent, which means it has been carefully considered and each context follows a deliberate order. It has been organised to ensure 4 key concepts are revisited over the primary age range; some year on year and some less frequently. As children learn through the curriculum, their conceptual understanding of computer components and resultant hardware, computational thinking, digital literacy and e-safety will become more secure as their knowledge and skills are enhanced through regular application. They will also revisit familiar opportunities over time, such as exploring animation in Year 2 and in Year 4.

Children will learn about computer components and how to operate computers safety and build functional skills, such as the use of the keyboard and navigation tools. As children grow older and develop a more altruistic view of the world, they will explore and learn about generic computer components to support their understanding of how components such as servers and cloud systems are designed and used within range of complex networks and systems around the world. Securing navigational skills, including confidence with keyboard word processing skills will support computational thinking programming tasks, as well as functional digital literacy skills often utilised in work and personal administrative tasks.

Central to developing the knowledge and skills as a computer scientist is computational thinking. Revisiting computational thinking year on year will be preferred as the children are less likely to use and apply this understanding beyond the school context.

As children age, the opportunities to explore digital technology will expand enabling them to experiment with sound, images and video, of which the tools and outcomes in this area will evolve as technology advances year on year. Children will have opportunities as they progress through the curriculum to learn about how digital content, such as animation, e-books, websites, are created and how they may create similar content. Opportunities such as keyboard skills, presentation skills, familiarising with data tools, and understanding of creative media e.g. animation, have all be chosen because of their transferrable skills and relevance to the world the children experience around them.

The key concept that technology should be used safely, respectfully, and responsibly refers not only to the safe handling of hardware, but the wider e-safety aspects of learning. E-safety is explicitly taught within the CLF Citizenship/PSHE curriculum. Opportunities to develop pupils e-safety are expected to be referred to within the teaching of Computer Scientists in local enactment of the curriculum.