SUBJECT: IT

CURRICULUM PLAN

I have come so that they may have life and have it to the full

John 10: Chapter 10. V 10.

Inspired by the Gospel values of love, joy and compassion, we nurture and develop the whole child so that each unique individual grows in confidence and has the knowledge and skills to enable them to achieve their true potential.

All children will have access to a rich, ambitious curriculum which fosters curiosity, creativity and a love of life-long learning. It will prepare them well for the next stage of education, employment or training and enable them, as educated citizens, to contribute to creating a better world.

Information Technology is delivered in the hope to inspire students who are interested in an introduction to the study of creating IT systems to manage and share information alongside other fields of study, with a view to progressing to a wide range of higher education courses, not necessarily in IT. Students will develop a common core of IT knowledge and study areas such as the relationship between hardware and software that form an IT system, managing and processing data to support business and using IT to communicate and share information. In addition to helping students acquire subject knowledge, the course is designed to develop the following skills:

- cognitive and problem-solving skills: use critical thinking, approach non-routine problems applying expert and creative solutions, use systems and technology
- Intrapersonal skills: communicating, working collaboratively, negotiating and influencing, self-presentation
- Interpersonal skills: self-management, adaptability and resilience, self-monitoring and development.

The National Curriculum

We have not deviated from the National Curriculum in this subject.

By the end of Key Stage 2

Pupils have studied:

- the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation.
- E-safety

They can:

- 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 technology purposefully to create, organise, store, manipulate and retrieve digital content.
- recognise common uses of information technology beyond school.
- 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 predict and 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; keeping personal information private; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact on the internet or other online technologies.

*Taken from Computing NC for KS1 and KS2
https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/239033/PRIMARY_national_curriculum_-_Computing.pdf

IT Department Catholic social teaching principles

Participation in society

The IT Department at St Edmunds Catholic School is dedicated to inspiring each and every student to achieve their academic potential. It also seeks to help young people develop life skills, through computing competency and helping them to become computer literate in order function in our ever-changing digital age. Students are encouraged to explore the internet to seek out information to help further comprehension in all subjects, and to develop insight as to what is a good resource and what is not so reliable. We will provide a broad and balanced curriculum that is accessible to all pupils. All students will have the opportunity to develop a high level of computer competency required for success in adult life. We will ensure that all pupils are able to thrive and develop as computer literate individuals.

Human Dignity

Digital Dignity pertains to the inherent right of individuals to maintain control over the digital identity, data, and online interactions, ensuring these are respected and protected within the digital sphere. Students have the opportunity and are given the knowledge to uphold their digital dignity.

Preferential option for the poor

We provide revision resources for all students, so they are not required to purchase revision guides. We also provide lunchtime, afterschool and holiday intervention so that no student is disadvantaged. We foreground the disadvantaged when teaching.

Solidarity

We offer the students opportunities to work in groups; this enables them to participate in discussions with their peers. We include activities such as think, pair share which encourage discussions with their peers, and builds teamwork skills.

The Common good

This refers to the application of IT to benefit society, often by promoting access to information, fostering collaboration, enabling democratic participation, and addressing public needs through digital platforms and technologies. IT can provide broad access to educational resources, transforming lives through accessible and impactful learning experiences.

Promoting Peace

Every student has the right to be in a safe and loving classroom environment where they can express their own opinion. It is the responsibility for the teacher to facilitate this. Every student has the right to a peaceful calm environment where they can express their thoughts without feeling unsafe and this is thoroughly promoted when we teach IT.

Subsidiarity

In IT students apply the principle of subsidiarity by making decisions, designing, and implementing those solutions. The teaching and homework provide ample opportunity for this so that all students become autonomous learners so that they become responsible for their learning.

Stewardship

Stewardship for IT means using technological gifts responsibly and ethically to serve the common good and the dignity of all people, including future generations. Our IT curriculum ensures IT benefits all, especially the vulnerable, by promoting justice and a sustainable future, rather than being used for exploitation, resource depletion, or environmental harm.

Distributive justice

The fair and equitable sharing of worldly resources, ensuring everyone's basic needs are met so they can achieve their full human potential. In IT we evaluate systems to see if they foster conditions for all to thrive, rather than allowing exploitation or the concentration of wealth and power.

SUBJECT KEY CONCEPTS / COMPOSITES	
Key Stage 3	• Computer Systems and Networks Digital devices Hardware and software CPU and the fetch-execute cycle Networking and the Internet Network security

	Cyber security Online safety • Programming and Computational Thinking Computational thinking Decomposition Pattern recognition Abstraction Algorithm design Searching algorithms Sorting algorithms Sequencing Selection Iteration Logical reasoning Programming using Turtle Python Programming using Python • Data and Digital Literacy Binary number systems Hexadecimal Representing text, images and sound Spreadsheets Data handling Data representation • Creating Digital Media Graphical design Digital media creation User-focused design Evaluating digital products • Responsible Use of Technology Online safety Digital footprints Bias and reliability

	Copyright Law and ethics
Key Stage 4	• Design tools: Types of design tools. • Human Computer Interfaces in everyday life: The purpose, importance and use of HCI in everyday life, The purpose, importance and use of HCI in application areas, Hardware considerations, Software considerations, User interaction methods. • Data and testing: Information and data, Data use, Use of data types in different contexts, Validation and verification, Data validation tools, Data verification tools, Data collection methods, Storage of collected data, Application of testing to a range of contexts, Importance and purpose of testing, Test data, Types of Testing. • Cyber security and legislation: Threats, The impact of a cyber security attack on individuals and organisations, Prevention Measures, Legislation related to the use of IT systems. • Digital communications: Types, Software, Digital devices, Distribution channels, Types of distribution channel, Distribution channel connectivity, Audience demographics: • Internet of Everything (IoE), Use of IoE , Application areas in everyday life.

CURRICULUM PLAN: KEY STAGE 3

	Year 7	Year 8	Year 9

Knowledge	***Pupils will know:***
	• How to access the school systems and use reliable sources for information
	  ○ Passwords and Accounts
	  ○ Drive and Classroom (File Storage)
	  ○ Email and Printing
	  ○ Google Research task
	• Programming essentials with Rapid Router
	  ○ Introduction to programming and sequencing
	  ○ Iteration
	  ○ Selection
	  ○ Variables
	  ○ Abstraction
	  ○ Subroutines
	• Computer systems and the history and future of computing
	  ○ History of computing
	  ○ Components of a computer
	  ○ Input and Output devices
	  ○ Types of software
	  ○ The Internet
	  ○ The future of computing
	• Online Safety
		Pupils will know:
		• Programming with Turtle Python
		  ○ Build on previously learned programming concepts
		  ○ Using an IDE to develop code
		  ○ Use of subroutines to draw shapes
		  ○ Predict, run, investigate, and modify code
		  ○ Iteration methods
		  ○ Using user inputs
		• Mathematics in computing
		  ○ Binary and Binary addition
		  ○ Hexadecimal numbers
		  ○ ASCII
		  ○ Image Representation
		  ○ Boolean logic
		• Hardware, Networks and the Internet
		  ○ The CPU and the Fetch, Execute and Decode cycle
		  ○ Primary and Secondary storage
		  ○ Networks and Topologies
		  ○ The Internet and the World Wide Web
		  ○ Network threats and Prevention measures
			Pupils will know:
			• Exploring Cyber Security
			  ○ Exploring online data and how it is used
			  ○ Social engineering
			  ○ Phishing
			  ○ Social Engineering techniques used by cyber criminals.
			  ○ Evaluate Ethical Hacking
			  ○ Understanding how Bots are used.
			  ○ Methods of preventing cyber attacks
			• Flowcharts
			  ○ Features of flowcharts
			  ○ Understanding software to produce flowcharts
			  ○ Converting flowcharts to code
			• Python programming
			  ○ Programming concepts: inputs, Strings, selection, loops and lists
			• Digital Media and 3D Design
			  ○ Keyframes
			  ○ Rendering
			  ○ Lighting
			  ○ Animation
			  ○ 3D models
			• Data Representation
			  ○ Binary

	• Programming essentials in Scratch Part 2 ○ Formalise the use of subroutines ○ Predict, run, investigate, and modify code ○ Evaluate the type of iteration required in a program ○ Introduction to lists ○ Create a translation quiz for a MFL teacher • Using media – Gaining support for a cause ○ Features of a word processor ○ Licensing appropriate images ○ Credibility of sources ○ Research and plan your blog ○ Promoting your cause ○ Project completion and assessment ***We have chosen to teach this knowledge because…*** Modern learners arrive with varied digital experiences, making it vital to establish an immediate, equitable foundation in school systems, safe communication, and online safety. Transitioning from block-based visual programming (Scratch) to early block-text hybrids (Rapid Router) develops	○ Encryption • Introduction to Spreadsheets ○ Understand the purpose of spreadsheets and where they are used ○ Identify rows, columns, cells and worksheets ○ Enter and edit data accurately ○ Use basic formulae to perform calculations ○ Perform addition, subtraction, multiplication and division using cell references ○ Use common spreadsheet functions including SUM, AVERAGE, MIN and MAX ○ Use COUNT and COUNTIF functions to analyse data ○ Format cells, tables and worksheets appropriately ○ Create charts and graphs to visualise data ○ Interpret spreadsheet data to identify trends and patterns • Introduction to Python programming ○ Write simple python programs. ○ Using simple arithmetic expressions to calculate values ○ Selection and randomness ○ Use if, Elif, else, and while statements ○ Use of iteration to control the flow of program execution	○ Images ○ Audio ○ Sampling ○ Compression • Modelling data - Spreadsheets ○ Getting to know a spreadsheet ○ Quick calculations ○ Collecting data ○ Become a data master! ○ Level up your data skills ***We have chosen to teach this knowledge because…*** This final KS3 milestone deepens technical competency and directly bridges the gap to advanced KS4 concepts like cyber security, design tools, and text-based coding. By exploring cyber security, data privacy, and ethical hacking, students learn how data is exploited online and how logical and physical prevention methods protect individuals and organisations. Introducing formal flowcharting teaches students to plan, visually map and evaluate algorithms before software development begins, providing a direct foundation for the design tools studied at Key Stage 4. Transitioning these flowcharts directly into syntax-driven Python programming

	computational thinking, helping students master core logic constructs—sequence, selection, and iteration—before tackling text-based code. Introducing networking fundamentals gives students a clear window into how data travels across the Internet, while exploring computer history and hardware components anchors their technical literacy. Finally, the digital media project balances technical skill with critical evaluation, teaching students to assess source credibility, understand copyright licensing, and use technology responsibly to advocate for others. This blend of safety, logic, and functional literacy ensures all students are equipped to participate safely and effectively in a digital society.	○ Combine iteration and selection • Media- Vector graphics ○ Draw and manipulate basic shapes ○ Using path operations to combine simple shapes into more complex ones ○ Convert objects to paths ○ Create a vector graphic design ○ Investigate vector graphics ***We have chosen to teach this knowledge because…*** As technology rapidly evolves, students must shift from being passive consumers of digital content to active creators and critical problem-solvers. Reintroducing programming through Turtle Python provides an engaging, visual bridge to formal text-based coding, allowing learners to apply mathematical concepts, logical structures and user input to real-world code execution. Exploring binary data representation, Boolean logic and the inner workings of the CPU through the Fetch-Decode-Execute cycle strips away the mystery of digital devices, helping learners understand how instructions are physically stored and processed. Introducing spreadsheets develops learners' ability to organise, analyse and interpret data using formulae, functions and charts, while also reinforcing	solidifies core constructs such as inputs, strings, selection, loops and lists, enabling students to debug errors and manipulate data structures. Finally, exploring 2D and 3D graphics suites allows learners to understand how spatial layouts, asset manipulation, and animations are engineered. This rigorous schema ensures students enter KS4 as analytically minded digital citizens with a secure baseline in secure systems and computational design.

		mathematical reasoning and problem-solving skills. Furthermore, balancing highly technical subjects with creative disciplines such as vector graphics provides learners with experience of real-world digital workflows, demonstrating how technology combines logical thinking with creativity and visual communication. This broad and balanced curriculum develops resilience, analytical thinking and digital confidence, providing a strong foundation for future study at Key Stage 4.	
Skills	***Pupils will be able to:*** • **Navigate and secure** the school network, cloud accounts, and learning platforms safely and autonomously using strong, unique passwords. • **Organise and maintain** structured file storage systems using appropriate folder structures and naming conventions for schoolwork. • **Draft and send** professional, respectful digital communications via email while identifying and avoiding cyberbullying behaviours. • **Design, build, and debug** block-based programs in Scratch and Rapid Router utilising core programming constructs: sequence, selection, iteration, and variables.	***Pupils will be able to:*** • **Transition from block to text-based code** by writing, running, and modifying functional Turtle Python programs to draw complex geometric shapes. • **Apply arithmetic operators** and simple arithmetic expressions to calculate values dynamically within Python scripts. • **Construct text-based control flows** combining multi-branch selection (if, elif, else) with condition-controlled iteration (while). • **Generate and deploy** random integers and relational operators to control dynamic program outcomes. • **Convert data** manually between binary numbers, hexadecimal	***Pupils will be able to:*** • **Formulate data privacy strategies** to minimise data exposure online, actively mitigating human error and social engineering risks. • **Analyse and differentiate** the motivations, threat profiles, and actions of different hacker types (such as white, grey, and black hat hackers). • **Recommend logical and physical security measures** to protect individual data and organisational networks against automated bots and cyber attacks. • **Construct and interpret algorithmic flowcharts** using standard terminal, process, decision, and input/output symbols to plan logical solutions.

	• **Decompose** simple algorithmic problems into logical subroutines to minimise code repetition. • **Differentiate** between internal computer hardware components, defining their specific input, output, or processing functions. • **Input, format, and filter** numerical datasets within a spreadsheet, executing basic cell calculations and generating appropriate charts. • **Plan and publish** a digital media project (blog) tailored to a specific target audience, ensuring all media complies with Creative Commons licensing rules. • **Critique online sources** for credibility, applying referencing techniques to avoid plagiarism.	systems, ASCII codes, and denary equivalents. • **Sketch and trace** basic logic circuits using AND, OR, and NOT logic gates to illustrate how hardware executes binary instructions. • **Explain and contrast** the difference between purpose-built and general-purpose hardware devices, including the function of the operating system. • **Design vector graphics** by manipulating basic shapes, editing path nodes, and applying path operations to combine elements.	• **Translate visual flowchart design documentation** directly into executable text-based Python code. • **Select and use specialised design software** to produce accurate, legible system flowcharts that meet a given brief. • **Write, execute, and debug text-based Python scripts** that integrate advanced variables, string formatting, and keyboard inputs. • **Control program execution flow** by nesting multi-branch selection statements (if-elif-else) inside iterative loops (while / for). • **Create and manipulate data structures** in Python by generating lists, accessing specific indices, and performing operations on data items. • **Model, rotate, and scale assets** accurately within a 3D application space to produce static graphics and multi-frame animations. • **Apply 2D design principles** alongside 3D rendering configurations to output digital media fit for a target audience.
Cultural Capital	***Pupils will all have been exposed to [the ideas of/the texts/the concepts/the composites]:*** • Cloud operating infrastructure, centralised network login	***Pupils will all have been exposed to [the ideas of/the texts/the concepts/the composites]:***	***Pupils will all have been exposed to [the ideas of/the texts/the concepts/the composites]:*** • Cyber security threats, automation bots, and data privacy vulnerabilities

	spaces, and secure password management. • Professional digital communication protocols, workplace email etiquette, and targeted cyberbullying reporting pathways. • Block-based computational logic, sequencing execution, and the hybrid block-text logic patterns within Rapid Router. • Core algorithmic control structures, including conditional branch selection and count-controlled loop iteration. • Dynamic variables, subroutine mapping, and decomposition frameworks inside visual scripting environments. • The foundational evolution of computer science history alongside modern physical internal computing hardware. • Distinctions between data and information, primary versus secondary data streams, and electronic folder management structures. • Functional spreadsheet design, lookup indices, condition-controlled processing (COUNTIF), and visual data analysis charts.	• Text-based code translation execution environments and Integrated Development Environments (IDEs). • Structural logic automation via Turtle Python and geometric shape rendering subroutines. • Multi-branch conditional selection logic chains (if-elif-else) and automated condition-controlled iteration (while loops). • Random integer parameters, mathematical arithmetic tracking, and relational data operations within textual scripts. • Binary number bases, base-16 hexadecimal structures, ASCII character string sets, and bitwise image matrix representations. • Fundamental processor architecture, system storage tiers, and the standard Fetch-Execute-Decode CPU sequence. • Network infrastructure nodes, topological layouts, global internet delivery channels, and data packet routing architectures. • Logical hardware network barriers (firewalls), system encryption models, and logic gate circuit blueprints (AND/OR/NOT). • Vector graphic creation, path node adjustments, geometric path	• Ethical hacking principles and defence methodologies • Flowchart design tools, standard symbol systems, and planning software • Code generation from structural design documentation • Text-based programming syntax, inputs, and string manipulation • Control structures including complex selection and conditional loops • Index-based data structures (Python lists) • 2D vector layouts and 3D modelling pipelines • Keyframe animation mechanics and digital asset rendering workflows ***This will enhance cultural capital because:*** Students develop an acute awareness of the legal boundaries, ethical responsibilities, and unseen mechanisms that govern our global digital infrastructure. Demystifying cyber security, structural engineering, and ethical hacking moves students past basic online safety rules, transforming them into proactive defenders of data privacy and individual rights within a heavily networked world. Learning how to plan systems using industry-standard

	• Word processor formatting standards, audience-focused typography layouts, and media project publishing platforms (blogs). • Intellectual property rules, Creative Commons copyright licenses, and digital content credibility evaluation matrices. ***This will enhance cultural capital because:*** Students shift from being passive, uncritical consumers of technology to responsible, informed participants in a digital society. By exploring the historical evolution of computer systems alongside current hardware, they gain an appreciation for the technological innovations that shape modern life. Mastering secure digital routines, formal communication etiquette, and critical online safety measures empowers them to navigate society safely, protect their personal digital dignity, and respect others. Furthermore, by learning to interrogate web content for bias and credibility, students develop the skills to resist misinformation. Finally, using digital media tools to research and publish a blog to advocate for a social cause instils a sense of agency, allowing	combination tools, and digital canvas asset workflows. ***This will enhance cultural capital because:*** Students develop a deeper understanding of how modern digital systems are designed, programmed and used to solve real-world problems. By studying binary representation, processor architecture and networking concepts, learners gain insight into the technologies that underpin everyday life. Text-based programming encourages logical thinking, resilience and precision, while spreadsheet modelling develops the analytical and data-handling skills used extensively in education, business and industry. The inclusion of vector graphics and digital design demonstrates how technology can be used creatively as well as technically. Together, these experiences provide learners with a broad understanding of the digital world and establish strong foundations for further study in Information Technology, Computing and related career pathways.	flowchart design tools and translating those visual blueprints into textual Python syntax simulates real-world software engineering practices. Finally, working within 2D layouts and 3D application spaces to build and render animations mirrors the exact pipelines used in modern media, gaming, and engineering sectors. This deep technical proficiency ensures that all students—regardless of background—graduate Key Stage 3 with the advanced digital literacy, critical thinking, and social awareness needed to confidently step onto competitive Level 2 pathways and future careers.

	them to see technology as a powerful vehicle for positive influence and social justice in the wider world.		
Assessment: Formative – classwork and homework. Summative – end of cycle assessments and end of year examination.	***Pupils will have been assessed on:*** • **Digital Safety and Infrastructure Alignment:** The ability to independently secure user accounts, configure high-strength passwords, navigate cloud file repositories, and report cyberbullying or e-safety anomalies. • **Algorithmic Construction and Logic:** Comprehension of core programming constructs (sequence, selection, count-controlled iteration, and variables) within Scratch and Rapid Router blocks. • **System Decomposition and Subroutines:** The ability to break down visual coding tasks into logical segments to optimise execution sequences. • **Hardware Literacy and Evolution:** Identification and categorisation of internal system architecture components alongside peripheral input, output, and storage devices. • **Data Handling and Analytical Modeling:** Accuracy in creating, formatting, filtering, and	***Pupils will have been assessed on:*** • **Textual Scripting and Execution Flow:** The syntax mechanics of text-based Python programming, incorporating arithmetic operators, math logic, inputs, and string assignment variables. • **Control Flow Structuring:** The programmatic integration of multi-branch conditional checking (if-elif-else) alongside continuous condition-controlled loops (while). • **Data Representation Systems:** Mathematical conversion proficiency across base-2 binary addition, base-16 hexadecimal indexing, ASCII characters, and pixel matrix assets. • **System Processing and Interconnectivity:** Comprehension of the processor Fetch-Execute cycle, operating system environments, hardware classifications, and logic circuit models (AND/OR/NOT logic gates). • **Network Architecture and Defense:** Explanation of topological mapping, data routing	***Pupils will have been assessed on:*** • **Threat Analysis and Threat Mitigation:** Identification and classification of cyber malware vectors, automation bots, hacker archetypes, and data exploitation risks to personal or organisational spaces. • **Regulatory and Statutory Compliance:** Application of digital legislation boundaries (Data Protection Act, Computer Misuse Act) to realistic scenarios involving data loss or system intrusion. • **Algorithmic Blueprinting and Conversion:** Technical design of detailed system flowcharts using standardised notation symbols to visually plan, map, and solve multi-tier programming logic. • **Design-to-Code Translation:** The syntax execution required to translate graphical flowchart layout plans directly into valid, functioning text-based Python environments. • **Advanced Data Structures and Arrays:** Structural management of dynamic variables, string

	performing basic calculation formulas (COUNTIF) within a functional spreadsheet grid. • **Audience-Centric Media and Compliance:** The structural formatting of a blog project, the execution of source validation criteria, and adherence to Creative Commons copyright frameworks. ***By…*** Pupils will complete online assessment forms and, also through project work.	protocols, packet architecture, and the logic behind encryption barriers and firewalls. • **Vector Precision and Path Manipulation:** Technical mastery of shape drawing tools, path operations, node adjustments, and scale manipulation to output clean 2D vector files. ***By…*** Pupils will complete online assessments through forms and, also through project work.	operations, and complex index lists within text-based conditional loops. • **Spatial Engineering and 3D Asset Mechanics:** The geometric creation, mesh rotation, scaling, extrusion, subdivision, and parenting of components inside an active 3D view pane. • **Rendering Automation and Lighting Layouts:** Setup configuration parameters involving virtual camera lines, light tracing, and gobo masking filters to export finished multi-frame animations. ***By…*** Pupils will complete online assessments through forms and, also through project work.

YEAR GROUP PLANS			
Year 7			
Cycle 1	**Cycle 2**	**Cycle 3**	**Cycle 4**
Pupils will know: • ***School System Architecture: How to access the school network, use Microsoft Office 365 Workspace, save files to personal drives, and use digital classroom portals.*** • ***Communication Infrastructure: How to send effective, polite, and safe emails (understanding recipients, subject lines, and professional text).*** • ***Digital Safety: The effects of cyberbullying, how to use privacy settings on social media, the risks of geotagging, and how to verify who they are talking to online.*** • ***Basic Digital Skills: How to follow rules in an IT suite and build baseline confidence navigating digital workstations.*** ***Pupils will be able to:***	***Pupils will know:*** • ***Network Fundamentals: What a basic computer network is, the definition of a protocol (rules that let devices talk to each other), and what bandwidth measures (data transmission speed).*** • ***Connectivity Hardware: The physical hardware elements used to link devices together on a network, including routers, hubs, connection ports, and network cables.*** • ***Wired vs. Wireless: The structural differences between physical wired loops and wireless delivery channels (such as Wi-Fi and Bluetooth parameters).*** • ***The Internet Grid: The core difference between the physical internet infrastructure and the web services that run on top of it, like the World Wide Web (WWW).*** ***Pupils will be able to:***	***Pupils will know:*** • ***Computational Execution: How computers process commands systematically using the linear input-process-output control flow loop.*** • ***Core Programming Logic: The rules of sequential programming (putting blocks or commands in a precise, logical order to achieve an outcome).*** • ***Dynamic Storage Constructs: The definition of a variable and how values are initialised and tracked over a basic program cycle.*** ***Pupils will be able to:*** • Use web browsers and search technologies effectively to locate specific online resources safely. • Predict and trace the outcome of a simple sequence of visual block commands. • Trace and update variable values step-by-step through	***Pupils will know:*** • ***Selection Logic: How to use basic conditional logic structures (like if/else statements) and comparative signs to guide a program's choices.*** • ***Iteration Frameworks: The definition of iteration, the engineering need for loops in computing, and when to use fixed count-controlled loops.*** • ***System Decomposition: The value of breaking a larger, complex computational problem down into smaller, manageable chunks or subroutines.*** • ***Document Design & Publishing: How to use core word processing text tools, formatting layout options, and online platforms like blogs to communicate ideas to an audience.*** ***Pupils will be able to:*** • Modify a program to include selection routines using

• Log in and navigate school platforms securely and track homework deadlines independently. • Organise digital workspaces by creating and using structured folders for schoolwork. • Draft and send respectful, formal emails to teachers or peers. • Identify and explain the signs and consequences of cyberbullying. • Apply basic safety checklists to check their own privacy profiles online. ***Pupils will all have been exposed to*** • ***Email efficacy protocols*** • ***Efficient folder organisation layouts*** • ***Personal data privacy checks*** • ***Different aspects of cyberbullying prevention*** ***Pupils will have been assessed on:*** • Summative online tests • Live classroom activity checks	• Define and list the key hardware components required to build a functioning local network. • Compare wired to wireless connections, explaining the benefits and limits of speed and range. • Explain the baseline journey of how data travels across the internet using IP addresses and packets. • Differentiate between the physical infrastructure of the internet and the World Wide Web service. ***Pupils will all have been exposed to:*** • *Computer network concepts and connectivity rules* • *Physical network hardware and connection ports* • *Wired and wireless transmission boundaries* • *The structural layers of the global internet and the World Wide Web* ***Pupils will have been assessed on:*** • *Visual network topology mapping diagrams* • *Direct teacher questioning and verbal checking arrays*	a straightforward programming line. • Build and run sequential scripts inside Scratch where blocks respond cleanly to manual user inputs. ***Pupils will all have been exposed to:*** • *World Wide Web search frameworks and browser layouts* • *Program input-process-output linear sequences* • *Mutable variables and runtime tracking tracking* • *Core programming sequence mechanics* ***Pupils will have been assessed on:*** • Sequential programming logic and computational thinking pathways. ***By:*** • Practical code-tracing worksheets and program predictions • Visual script building tasks within Scratch • Active recall classroom starter quizzes • Active Recall • Classroom Activities/projects • Homework tasks	simple true or false logic triggers. • Implement count-controlled loops in Scratch to create repeating movement or animation sequences without duplicate code blocks. • Detect and fix simple errors (debugging) within their own visual script passages. • Decompose a basic task into separate subroutines to organise code efficiently. • Use word processors and blog applications to style and format a clean content layout tailored to an audience. ***Pupils will all have been exposed to:*** • *Selection statements and logical comparative tools* • *Count-controlled iteration loop parameters* • *System decomposition methods and subroutine usage* • *Word processor text layouts and audience-facing blog formats* ***Pupils will have been assessed on***

• Low-stakes retrieval homework tasks • Active Recall • Classroom Activities/projects • Homework tasks	• *Short retrieval knowledge checks*		• Selection, count-controlled iteration, and digital document construction workflows. ***By:*** • End-of-cycle online written assessments • Completed block-coding functional scripts • Practical blog writing project file submissions • Active Recall • Classroom Activities/projects • Homework tasks
Literacy ***Key terms:*** • **Password:** A secret string of characters used to authenticate access to a digital account. • **Workstation:** A single desktop computer or computer setup connected to a local network. • **Cloud storage:** Remote servers accessed via the internet to store and manage user data. • **Digital footprint:** The permanent record of data	*Literacy* ***Key terms:*** • **Data:** Unprocessed, raw facts, symbols, or statistics without situational context. • **Information:** Data that has been processed, structured, and given clear meaning. • **Network:** A collection of interconnected computer systems sharing hardware and resources. • **Protocol:** A strict set of rules governing data transmission between digital devices. • **Bandwidth:** The maximum capacity and rate of data	*Literacy* ***Key terms:*** • **Search engine:** A web-based program that indexes and searches database records using keywords. • **URL:** Uniform Resource Locator; the specific global address of a web resource. • **HTTP/HTTPS:** Protocols defining how web page data is requested and securely transmitted. • **Sequence:** A core programming construct where instructions execute in a precise linear order.	*Literacy* ***Key terms:*** • **Selection:** A construct where a program chooses an execution path based on a condition. • **Iteration:** A core programming loop that repeats a sequence of instructions multiple times. • **Decomposition:** Breaking down a complex computational problem into smaller, manageable subproblems. • **Subroutine:** A self-contained, named block of

left behind by an individual's online activities. • **Cyberbullying:** The use of electronic communication to bully, intimidate, or harass a person. • **Geotagging:** Attaching geographical coordinates and location data to digital media files. • **Privacy settings:** Controls on a platform that restrict who can view user information and content. ***Specific Vocabulary:*** • Communication: The systematic imparting, sharing, or exchanging of digital information. • Recipient: The specific individual or system designated to receive an incoming message. • Authenticate: Verifying the true identity of a user or process attempting system access. • Infrastructure: The underlying physical and	transmission over a connection. • **Router:** A network hardware device that directs data packets between different networks. • **Internet:** The global physical infrastructure of interconnected computer networks. • **World Wide Web:** A system of internet servers supporting hyperlinked multimedia documents. ***Specific Vocabulary:*** • **Transmission:** The physical action or process of moving data packets across a network. • **Connectivity:** The technical capacity of independent devices to link and communicate. • **Topology:** The structural layout arrangement of devices within a network framework. • **Primary source:** Original, first-hand data collected	• **Variable:** A named memory storage location used to hold data that can change during execution. • **Input-Process-Output:** The fundamental processing loop that controls all computer systems. ***Specific Vocabulary:*** • **Execution:** The operational process of a computer system carrying out programmatic commands. • **Symmetrical:** Formed of exactly similar parts facing each other across an axis. • **Compilation:** The translation process converting human-readable instructions into executable code. • **Mutable:** Subject to alteration or change during a program's active runtime. • **Syntax:** The specific structural rules governing the arrangement of symbols and programming terms.	code that performs a specific operational task. • **Debugging:** The technical process of identifying, isolating, and fixing code errors. • **Word processor:** Software application used for the creation, editing, and formatting of text documents. • **Blog:** A regularly updated website or platform managed to communicate ideas to an audience. ***Specific Vocabulary:*** • Conditional: Depended on a specified true or false state being met before executing. • Modular: Composed of distinct, independent units that combine to form a larger system. • Typography: The structural art and technique of arranging type to make text legible. • Hierarchy: An organisational layout framework that ranks

software systems supporting digital school environments. • Vulnerability: A weakness or security flaw within a system that exposes personal data.	directly for a specific investigative purpose. • **Secondary source:** Existing information or research data compiled previously by a third party.		visual elements by order of importance. • Formatting: The mechanical arrangement and stylistic design of text and assets within a document.
Connections ***We have identified these connections to other subject areas:*** • **PSHE:** Deep exploration of online safety, digital well-being, social responsibility, and the emotional impacts of cyberbullying. • **English:** Application of formal writing styles, professional communication etiquette, and appropriate tone for digital audience engagement. • **Executive Skills:** Functional management of cloud files, directory structures, digital planner tracking, and deadline monitoring workflows.	*Connections* ***We have identified these connections to other subject areas:*** • **Geography:** Investigation of global infrastructure grids, physical connectivity links, and the physical scaling of worldwide communication systems. • **Science:** Comparing the flow of data packets over network paths to physical distribution systems and biological transmission loops. • **History:** Analysing structural methods of long-distance communication and how systemic rule sets (protocols) have shaped human trade and infrastructure.	*Connections* ***We have identified these connections to other subject areas:*** • **Mathematics:** Practical application of comparative logic structures, tracking dynamic unknown values (variables), and step-by-step logic loops. • **Modern Foreign Languages (MFL):** Structuring translation commands, checking linguistic output sequences, and analysing the strict structural syntax of computer code. • **Design Technology:** Execution of linear systemic planning workflows through input, process, and output control loops.	*Connections* ***We have identified these connections to other subject areas:*** • **Art and Design:** Implementation of professional layout theory, typographic hierarchy balances, and visual balance within digital content spaces. • **English:** Planning structured longer-form written blogs, editing descriptive accounts, and applying precise academic source attribution. • **Citizenship:** Understanding user legal protections, global intellectual property criteria, and ethical creative asset licensing boundaries.

Learning at home ***Consolidation of learning will take place at home by…*** • Completing online homework and/or sheets which run concurrently with the lesson structure.	*Learning at home* ***Consolidation of learning will take place at home by…*** • Completing online homework, and/or sheets which run concurrently with the lesson structure.	*Learning at home* ***Consolidation of learning will take place at home by…*** • Completing online homework, and/or sheets which run concurrently with the lesson structure.	*Learning at home* ***Consolidation of learning will take place at home by…*** • Completing homework, and/or sheets which run concurrently with the lesson structure.

Sequencing

The Year 7 curriculum is intentionally sequenced to establish an immediate, equitable baseline in digital literacy and workspace security, recognising that pupils arrive with highly varied computing experiences from Key Stage 2. Learners begin by mastering safe school system architecture and professional digital communication protocols, ensuring they can seamlessly access, track, and manage their academic data across Key Stage 3 and Key Stage 4. This early focus directly reinforces vital online safety guidelines, media privacy settings, and a critical awareness of digital footprints in a safe environment.

The curriculum then transitions logically from system access to foundational network infrastructure, exploring how data packets and protocols drive the global internet. This contextual knowledge provides a strong conceptual bridge to programming, where pupils shift from passive consumers to active logical thinkers. By investigating the input-process-output loop, learners develop computational thinking skills through sequential script composition and variable tracking within visual environments.

Finally, the year concludes by layering conditional selection and count-controlled iteration onto these visual logic foundations, alongside professional text document design. This deliberate trajectory ensures pupils build a secure vocabulary and resilient problem-solving mindset, preparing them to progress confidently to Year 8, where they will revisit online safety rules, explore raw text-based languages like Python, and study physical logic gates.

YEAR GROUP PLANS			
Year 8			
Cycle 1	**Cycle 2**	**Cycle 3**	**Cycle 4**
Pupils will know: • ***Text-Based Scripting (Introduction to Python): Writing simple Python programs, using accurate syntax rules, and navigating code files.*** • ***Mathematical Logic in Code: Using simple arithmetic expressions to calculate values dynamically inside text-based programs.*** • ***Interactive Software Mechanics: Receiving manual input from the keyboard to convert to a numerical value for system processing.*** • ***Digital Literacy & Plagiarism Checks: Applying referencing techniques to avoid plagiarism when handling research data and designing layouts (such as blogs) suitable for a target audience.*** ***Pupils will be able to:*** • ***Write and run simple Python programs using***	***Pupils will know***: • ***Advanced Python Flows: The programmatic control parameters of selection, randomness, and continuous loop systems.*** • ***Multi-Branch Coding Checks: How to use conditional logic statements (if, elif, else) to switch program execution paths.*** • ***Textual Iteration Patterns: How while loop statements control the continuous flow of program execution based on logic rules.*** • ***Complex Logic Control: Methods to combine selection logic inside iterative loops to handle advanced computational problems.*** ***Pupils will be able to:*** • ***Incorporate random integer triggers inside a script to automate random software selections.*** • ***Construct multi-branch conditional flows using valid if-elif-else code strings.***	***Pupils will know***: • Data Representation Maths: The mechanical rule that computing data and software instructions must be processed using binary bases. • Base System Architecture: The conversion rules for 8-bit binary structures, hexadecimal numbering indices, ASCII text mapping codes, and pixel image representation matrices. • Hardware Infrastructure Logic: The engineering purpose of the CPU, the steps of the Fetch-Decode-Execute cycle, and the distinction between primary and secondary system storage. • The Internet Grid: Network routing frameworks, network topologies, the physical infrastructure of the internet vs. the World Wide Web service, security threats, preventions, and encryption steps. ***Pupils will be able to:***	***Pupils will know***: • ***Textual Iteration Patterns: How while loop statements control the continuous flow of program execution based on logical parameters.*** • ***Complex Flow Control: Methods to combine selection logic inside iterative loops to control and handle advanced text programs.*** • ***Digital Media Mechanics: What vector graphics are and how they differ from pixel-dependent layouts.*** • ***Vector Precision Rendering: How complex digital art designs are constructed by manipulating groups of objects, combining paths, and editing path nodes.*** ***Pupils will be able to:*** • Use iteration loop structures (while loops) to smoothly control a program's execution.

text characters instead of visual blocks. • ***Locate and correct common syntax errors inside simple coding scripts.*** • ***Construct arithmetic expressions in Python to perform quick system calculations.*** • ***Gather text data inputs from a keyboard and safely cast them to numerical variables.*** • ***Design a clean layout structure for an audience-centric data document or blog while attributing sources correctly.*** ***Pupils will all have been exposed to:*** • *Text-based programming language syntax* • *Arithmetic variables and assignment statements* • *Standard keyboard user input capturing* • *Software tools to make a blog layout* • *Plagiarism prevention checking procedures* ***Pupils will have been assessed on***	• ***Deploy condition-controlled iteration statements (while) to repeat code execution smoothly.*** • ***Combine loop syntax with nested selection checks to manage how text programs operate.*** ***Pupils will all have been exposed to:*** • Relational data comparison operators • Binary and multi-branch selection parameters • Condition-controlled iteration methods • Automated pseudo-random integer generation • Nested programming control patterns ***Pupils will have been assessed on:*** • **Python control flows, loop syntax, and condition checks**. ***by***: • Summative practical text programming scenario checks • Code prediction worksheets and syntax tracking logs	• Convert data values manually between binary numbers, hexadecimal codes, and standard denary characters. • Perform manual binary addition calculations over standard 8-bit data registers. • Trace how data instructions navigate CPU components during the fetch-decode-execute cycle. • Differentiate between primary and secondary storage hardware solutions. • Map local network topographies and outline logical prevention measures used to encrypt data packets against network threats. ***Pupils will all have been exposed to:*** • ***Binary addition columns*** • ***Hexadecimal number conversions*** • ***ASCII data character encoding sets*** • ***CPU fetch-decode-execute stages*** • ***Primary versus secondary memory tiers*** • ***Network topologies, encryption keys, and logical barriers***	• Combine loops with nested selection checks to manage how text scripts run. • Draw, group, and manipulate basic shapes accurately inside a vector graphic design workspace. • Convert standard visual objects to paths and edit path nodes to reshape a vector graphic asset. • Combine multiple tools to create a finished vector graphic design. ***Pupils will all have been exposed to:*** • ***Condition-controlled iteration rules (while loops)*** • ***Nested programming control patterns (combining loop and selection syntax)*** • ***Vector graphic properties and path layouts* Object grouping and path transformation tools*** • ***Path node modification and editing*** ***Pupils will have been assessed on:*** • **Python control flows and vector graphic design manipulation** workflows.

• **Introduction to Python scripting** and baseline digital literacy application rules. ***By:*** • Completed practical Python calculation files • Low-stakes retrieval homework tasks • Active Recall • Classroom Activities/projects • Homework tasks	• Active recall classroom starter quizzes • Classroom Activities/projects • Homework tasks	***Pupils will have been assessed on:*** • **Computing systems processing rules, networking, and data representation** conversions. ***by***: • Summative online test • Verbal feedback from teacher • Active Recall • Classroom Activities/projects • Homework tasks	***by***: • Summative online test • Verbal feedback from teacher • Active Recall • Classroom Activities/projects • Homework tasks
Literacy ***Key terms:*** • **Syntax:** The precise structural rules governing the arrangement of symbols and text characters inside code files. • **Arithmetic expression:** A programmatic statement using mathematical signs to calculate numbers during system execution. • **Assignment statement:** A line of code that stores a specific value or computational outcome inside a named variable.	***Literacy*** ***Key terms:*** • **Selection:** A control construct where a program chooses a branch path based on a logical condition match. • **Multi-branch selection:** A logical chain using if, elif, and else to check multiple conditions in order. • **Randomness:** Generating unexpected, unpredictable pseudo-random numbers within a software script.	***Literacy*** ***Key terms:*** • **Binary:** A base-2 numbering scheme using only digits 0 and 1 to store and process computer data. • **Hexadecimal:** A base-16 tracking system that simplifies long binary files into compact codes. • **ASCII:** American Standard Code for Information Interchange; a system linking binary patterns to text symbols. • **Image representation:** How screen images are mapped	***Literacy*** ***Key terms:*** • **Vector graphic:** Digital imagery constructed via mathematical path coordinates, enabling perfect resizing without pixelation. • **Path node:** An anchor point on a vector path line that dictates its position, curvature, and direction. • **Object grouping:** Combining individual visual shapes or paths together so they can be manipulated as a single asset.

• **Keyboard input:** Data captured directly from a user typing on a physical keyboard device. • **Blog layout:** The visual arrangement of headers, text blocks, and media panels designed to communicate content to an audience. • **Plagiarism:** The act of copying another person's code or documentation text without proper source citation. ***Specific Vocabulary:*** • ***Semantics: The underlying meaning or operational logic of a specific programming statement.*** • ***Numerical value: Data consisting strictly of numbers that can be used in mathematical calculations.*** • ***Referencing: The formal process of attributing original research data or***	• **Iteration:** A core programming loop that repeats a sequence of software instructions multiple times. • **While loop:** A condition-controlled loop that continues to repeat code as long as a specific rule remains true. ***Specific Vocabulary:*** • **Relational operator:** Symbols used to compare values, such as greater than (>), less than (<), or equal to (==). • **Binary selection:** A straightforward code choice with only two possible paths (true or false). • **Nesting:** Placing one programming control statement directly inside another (e.g., selection checks inside loops). • **Counter variable:** A specialized variable used to track and count the number of times a loop executes.	out digitally as a matrix grid of pixels and bits. • **CPU:** Central Processing Unit; the primary hardware chip that executes software instruction sequences. • **Fetch-Decode-Execute:** The essential sequence loop executed by a CPU to process data instructions. • **Primary storage:** Fast, temporary memory accessed directly by the CPU, such as RAM or Cache. • **Secondary storage:** Permanent, non-volatile storage hardware used to keep files safe, such as HDDs and SSDs. • **Topology:** The structural arrangement or layout map of devices linked together within a network. • **Encryption:** The protective process of converting data strings into a secure format to stop network packet interception. ***Specific Vocabulary:***	• **Spreadsheet:** Software composed of cell grids used to organise, track, calculate, and analyse data. • **Quick calculation:** Utilizing shorthand basic formulas to sum, subtract, or average numerical blocks instantly. ***Specific Vocabulary:*** • **Path operation:** Technical commands used to unite, intersect, or exclude simple vector shapes to make complex layouts. • **Rasterisation:** The process of converting a mathematically defined vector asset into a pixel-dependent grid file. • **Data cleansing:** Identifying and fixing inaccurate, incomplete, or corrupted records inside a collected spreadsheet dataset. • **Cell referencing:** Pointing a spreadsheet formula to a specific grid coordinate (e.g., B4) to extract its active value. • **Data visualization:** Transforming raw numerical

images back to their true creator. • ***Layout template: A structural blueprint used to organise visual assets cleanly for a target audience.*** • ***Casting: Converting a data variable from one form to another (such as turning input text into a numerical integer).***	• **Flag variable:** A Boolean variable used as a signal to turn an execution flow on or off.	• Volatile: A temporary storage condition where data is wiped instantly if system power drops out (e.g., RAM). • Data register: A tiny, super-fast temporary storage pocket located directly inside the CPU. • Overflow error: A system error that occurs when a calculation produces a number too large for the data register to hold. • Threat vector: A specific path or method used by a cyber threat to gain unauthorised access to a network. • Packet routing: The mechanism of breaking data into small pieces and directing them across a network infrastructure.	cells into clear charts or graphs to reveal tracking patterns.
<u>Connections</u> *We have identified these connections to other subject areas:* • **English:** Applying logical rules to script layouts, matching the precision needed for punctuation and grammar with the	*<u>Connections</u>* *We have identified these connections to other subject areas:* • **Mathematics:** Practical execution of logical comparisons, working with variable values, and understanding how data	*<u>Connections</u>* *We have identified these connections to other subject areas:* • *Mathematics: Performing column addition calculations within different number bases, such as base-2 binary strings, and working with*	*<u>Connections</u>* *We have identified these connections to other subject areas:* • Mathematics: Practical application of coordinate planes, handling scale parameters, and editing vector path node lines using spatial tracking geometry.

strict syntax of Python code writing. • **Media Studies:** Planning document layouts and blog styles, ensuring fonts, spaces, and image choices are tailored to a specific audience. • **Business Studies:** Understanding the importance of original content creation, protecting creative ideas, and learning about patents and intellectual property.	flow charts branch based on numerical equations. • **Science:** Linking computer control loops and selection scripts to the way structured rule sets and automated commands guide real-world laboratory equipment. • **Modern Foreign Languages (MFL):** Translating abstract thought into structured command syntax, noting how small changes in vocabulary symbols shift the overall meaning of a line.	*base-16 hexadecimal indexing.* • *Science: Investigating electrical circuit flows through physical logic gate states, and exploring the physics behind magnetic disks versus solid-state storage chips.* • *History: Exploring the timeline of encryption codes and evaluating how secure messaging systems have altered historical events and network communication routes.*	• Art and Design: Working with digital canvas tools to draw shapes, combine overlapping geometric lines, and group visual assets into high-quality graphic designs. • Business Studies: Organising, tracking, and filtering user datasets inside cell grids to support rapid calculations and data-driven company decision-making.
<u>Learning at home</u> Consolidation of learning will take place at home by: • Completing homework online, and/or sheets which run concurrently with the lesson structure.	*<u>Learning at home</u>* *Consolidation of learning will take place at home by…* • Completing homework online, and/or sheets which run concurrently with the lesson structure.	*<u>Learning at home</u>* *Consolidation of learning will take place at home by…* • Completing homework online, and/or sheets which run concurrently with the lesson structure.	*<u>Learning at home</u>* *Consolidation of learning will take place at home by…* • Completing homework, sheets which run concurrently with the lesson structure.
<u>Sequencing</u> The Year 8 curriculum is carefully sequenced to transition pupils from visual logic into formal, text-based programming environments and to unmask the internal processing mechanics of modern computing systems. Building upon the core input-process-output foundations and safe user routines established in Year 7, Year 8 immediately introduces text-based Python scripting in Cycles 1 and 2. This early placement allows learners to build logical			

resilience as they learn to navigate a text-based environment, capture keyboard inputs, track simple arithmetic variables, and structure multi-branch selection flows using if-elif-else and while statements.

Once pupils become confident with raw code execution, the curriculum drops into the mathematical foundations of computing systems in Cycle 3. Unpacking bitwise data representation including binary number conversions, binary addition, hexadecimal indexing, ASCII characters, and pixel image matrices, shows learners how hardware physically interprets data. This maps directly into the study of computer systems, where students trace how these binary instructions flow through the CPU (the Fetch-Decode-Execute cycle) and navigate hardware network topologies, encryption paths, and internet infrastructure threats.

Finally, the year concludes in Cycle 4 by matching technical logic with creative design tools and business data modelling. By mastering mathematical vector graphic path node manipulation and learning to organize, track, and process data sets inside analytical spreadsheets, students explore real-world digital workflows. This deliberate trajectory ensures students build the advanced computational baseline and technical independence required to confidently face the data science tracking, complex system flowcharting, text-based arrays, and 3D modelling demands of Year 9.

YEAR GROUP PLANS			
Year 9			
Cycle 1	**Cycle 2**	**Cycle 3**	**Cycle 4**
Pupils will know: Cyber Security • The difference between data and information • Data privacy and personal data • Human error and security risks	***Pupils will know***: • Digital Media and 3D Design • Principles of digital media • 2D and 3D graphics • Dimensions and coordinates • Scale and proportions • Animation principles	***Pupils will know***: Flowcharts • Purpose of flowcharts • Standard symbols • Input/Output symbols • Decision symbols • Processes	***Pupils will know***: Data Representation • Binary numbers • Images • Pixels • Resolution • Colour depth • File size calculations

• Social engineering • Phishing attacks • Malware threats • DDoS attacks • Ethical hacking • Security prevention measures • Data Protection Act Data Science • What Data Science is • Data collection • Data cleansing • Data visualisation • Correlation • Outliers • Trends and patterns • Investigative cycle ***Pupils will be able to:*** • ***Explain cyber security threats and risks.*** • ***Assess methods used to protect data.*** • ***Evaluate the impact of cyber attacks.*** • ***Explain the importance of data privacy.*** • ***Collect and organise data for investigation.*** • ***Use software tools to visualise data.*** • ***Analyse data to identify trends and patterns.***	• Keyframes • Rendering • Lighting • Camera positioning • Asset design • Audience considerations • Digital media workflows ***Pupils will be able to:*** • ***Create 3D models*** • ***Create animations*** • ***Apply textures and materials*** • ***Use keyframe animation*** • ***Evaluate media products*** • ***Design assets for a given audience*** ***Pupils will all have been exposed to:*** • ***3D modelling software*** • ***Animation principles*** • ***Keyframing*** • ***Object scaling*** • ***Object rotation*** • ***Materials and textures*** • ***Proportional editing*** • ***Subdivision*** • ***Face editing*** • ***Extrusion*** • ***Parenting*** • ***Camera positioning*** • ***Lighting techniques*** • ***Rendering processes***	• Connectors • Advantages and disadvantages • Industry uses Python programming • Variables • Inputs • Outputs • String manipulation • Selection • Iteration • Lists • Functions • Debugging ***Pupils will be able to:*** • ***Create flowcharts*** • ***Interpret flowcharts*** • ***Convert flowcharts to code*** • ***Develop Python solutions*** • ***Debug Python programs*** • ***Use lists effectively*** ***Pupils will all have been exposed to:*** • Flowcharts • Standard flowchart symbols • Process symbols	• Audio representation • Sampling rate • Sample resolution • Compression Spreadsheet modelling • Spreadsheet structure • Cells • Rows • Columns • Formulae • Functions • Data validation • Charts • Visualisation • Data analysis ***Pupils will be able to:*** • Calculate image file sizes • Calculate audio file sizes • Edit images • Manipulate audio • Create spreadsheets • Use formulae • Create charts • Analyse data ***Pupils will all have been exposed to:*** Data Representation • ***Binary representation***

• ***Apply the investigative cycle to solve problems.*** • ***Present findings using appropriate visualisation methods.*** • ***Justify conclusions using evidence from data.*** ***Pupils will all have been exposed to:*** • ***Data and information*** • ***Data privacy*** • ***Human error*** • ***Social engineering*** • ***Phishing*** • ***Malware*** • ***DDoS attacks*** • ***Ethical hacking*** • ***Cyber security prevention measures*** • ***Data Protection Act*** • ***Data collection*** • ***Data cleansing*** • ***Data visualisation*** • ***Correlation*** • ***Outliers*** • ***Trends*** • ***Investigative cycle*** ***Pupils will have been assessed on*** • Understanding of cyber security threats and risks. • Knowledge of data protection and privacy.	• ***Digital media workflows*** • ***Asset creation*** • ***Audience-centred design*** • ***Animation sequencing*** • ***Spatial design principles*** ***Pupils will have been assessed on:*** • ***Understanding of animation principles and workflows.*** • ***Ability to manipulate and transform 3D objects.*** • ***Application of modelling tools and techniques.*** • ***Use of keyframes to create animations.*** • ***Selection and application of materials and textures.*** • ***Ability to create and organise animation assets.*** • ***Understanding of rendering and lighting techniques.*** • ***Evaluation of completed digital media products.*** • ***Application of appropriate technical terminology.*** • ***Production of a completed animation project.*** ***by***: • Summative online test • Verbal feedback from teacher • Active Recall • Classroom Activities/projects	• Decision symbols • Input/Output symbols • Terminators • Algorithms • Design documentation • Python programming • Variables • String manipulation • Inputs and outputs • Selection • Iteration • Lists • Functions • Debugging • Software development processes • Flowchart-to-code conversion ***Pupils will have been assessed on:*** • ***Understanding of flowchart conventions and symbols.*** • ***Ability to interpret and analyse flowcharts.*** • ***Ability to create flowcharts to solve problems.*** • ***Ability to convert flowcharts into working code.***	• ***Pixels*** • ***Resolution*** • ***Colour depth*** • ***Image file sizes*** • ***Image editing*** • ***Sampling frequency*** • ***Sample resolution*** • ***Audio representation*** • ***Compression*** • ***Digitisation*** • ***Bitmap graphics*** • ***Audio editing*** Spreadsheet Modelling • ***Spreadsheet software*** • ***Data collection*** • ***Data cleansing*** • ***Formulae*** • ***Functions*** • ***Cell references*** • ***Data validation*** • ***Charts*** • ***Graphs*** • ***Data visualisation*** • ***Data analysis*** • ***Modelling data*** ***Pupils will have been assessed on:*** • ***Understanding of image representation.*** • ***Understanding of audio representation.***

• Application of cyber security prevention measures. • Ability to identify and explain social engineering techniques. • Understanding of malware and network threats. • Ability to interpret and analyse data. • Use of visualisation tools to identify trends and patterns. • Understanding and application of the investigative cycle. • Evaluation of data sets and conclusions drawn from findings. ***By:*** • Summative online test • Verbal feedback from teacher • Active Recall • Classroom Activities/projects • Homework tasks	• Homework tasks	• ***Understanding of Python programming constructs.*** • ***Use of variables, inputs and outputs.*** • ***Use of selection and iteration.*** • ***Manipulation of lists and data structures.*** • ***Identification and correction of program errors.*** • ***Application of computational thinking and problem-solving skills.*** ***by***: • Creation and Evaluation of their independent project • Summative online test • Verbal feedback from teacher • Active Recall • Classroom Activities/projects • Homework tasks	• ***Calculation of image and audio file sizes.*** • ***Ability to explain how digital images are stored.*** • ***Ability to explain how digital audio is stored.*** • ***Application of image editing techniques.*** • ***Understanding of spreadsheet structures and terminology.*** • ***Use of formulae and functions.*** • ***Ability to collect, organise and analyse data.*** • ***Production of charts and visualisations.*** • ***Ability to interpret data and identify patterns and trends.*** • ***Application of spreadsheet modelling techniques to solve problems.*** ***by***: • Summative online assessment • Spreadsheet modelling project • Data representation calculations

			• Creation of charts and visualisations • Verbal feedback from teacher • Active Recall • Classroom Activities/projects • Homework tasks
Literacy ***Key terms*** Data – Raw facts and figures that have not yet been processed. Information – Data that has been processed into a meaningful form. Cyber Security – The protection of digital systems, networks and data from attack. Malware – Malicious software designed to damage or gain unauthorised access to systems. Phishing – A fraudulent attempt to obtain sensitive information through deception. Social Engineering – Manipulating people into revealing confidential information. DDoS Attack – A Distributed Denial of Service attack that overwhelms a system with traffic. Data Privacy – The protection and secure handling of personal information.	Literacy Key terms: • Animation – The process of creating the illusion of movement using a sequence of images. • Keyframe – A frame marking the beginning or end of a transition. • Render – The process of generating a final image or animation from a model. • Scale – Changing the size of an object. • Rotation – Turning an object around an axis. • Extrude – Extending a 2D shape into a 3D form. • Subdivision – A modelling technique used to create smoother surfaces.	Literacy Key terms: • Flowchart – A diagram used to represent the steps in a process or algorithm. • Algorithm – A sequence of instructions designed to solve a problem. • Process – A step that performs an operation within a flowchart. • Decision – A point where a choice is made based on a condition. • Input – Data entered into a system. • Output – Information produced by a system. • Variable – A named storage location for data. • Selection – A programming construct	Literacy Key terms: • Binary – A number system using only 0 and 1. • Pixel – The smallest element of a digital image. • Resolution – The number of pixels used to create an image. • Colour Depth – The number of bits used to represent the colour of a pixel. • Sampling Frequency – The number of times per second an audio signal is sampled. • Sample Resolution – The number of bits used to store each audio sample. • Compression – Reducing the size of a file.

Correlation – A relationship between two sets of data. Outlier – A data value that differs significantly from other values in a data set. Data Visualisation – Presenting data graphically to help identify patterns and trends. Data Cleansing – Identifying and correcting inaccurate or incomplete data. ***Specific Vocabulary:*** • Ethical Hacking – Authorised testing of systems to identify security weaknesses. • Threat – A potential danger to a computer system or network. • Vulnerability – A weakness that could be exploited by a security threat. • Mitigation – Measures taken to reduce the risk of an attack. • Investigative Cycle – A structured process used to collect, analyse and draw conclusions from data. • Trend – A general direction or pattern shown by data over time.	• Texture – A surface image applied to a 3D object. • Material – Properties that determine how an object appears when rendered. • Lighting – Digital light sources used to illuminate a scene. • Camera – A virtual viewpoint used to capture a rendered image or animation. • Parenting – Linking objects so they move together. Specific Vocabulary: • Mesh – The collection of vertices, edges and faces making up a 3D object. • Vertex – A point that forms part of a 3D model. • Scene – The complete 3D workspace containing objects, lighting and cameras. • Timeline – The sequence of frames used in animation. • Asset – A digital object used within a project. • Wireframe Model – A basic structural view showing the framework of a 3D object.	that chooses between alternative paths. • Iteration – Repeating instructions within a program. • List – A collection of multiple data items stored together. • Debugging – Identifying and correcting errors in code. • Function – A reusable block of code that performs a specific task. Specific Vocabulary: • Syntax – The rules that govern the structure of code. • Logic – The reasoning used to determine program behaviour. • Boolean – A data type with two values: True or False. • Condition – A statement that evaluates to True or False. • Parameter – A value passed into a function. • Pseudocode – A structured description of	• Spreadsheet – Software used to organise, analyse and calculate data. • Formula – An expression used to perform calculations. • Function – A predefined formula that performs a specific calculation. • Cell Reference – The location of a cell in a spreadsheet. • Data Visualisation – Presenting data using charts and graphs. Specific Vocabulary: • Bitmap Image – An image made up of individual pixels. • Vector Graphic – An image represented using mathematical shapes rather than pixels. • Digitisation – Converting analogue information into digital form. • Sampling – Measuring an analogue signal at regular intervals.

		an algorithm written in plain language.	• Data Validation – Checking that entered data is sensible and reasonable. • Chart – A graphical representation of data. • Cell – The intersection of a row and column within a spreadsheet. • Modelling – Using spreadsheets to represent and investigate scenarios.
<u>Connections</u> We have identified these connections to other subject areas: • Mathematics – Interpreting graphs, analysing trends, identifying correlations and outliers, and drawing conclusions from data. • Business Studies – Data protection, cyber security, organisational data handling and decision-making using data. • Citizenship – Online responsibility, digital identity, privacy and ethical use of technology.	*<u>Connections</u>* *We have identified these connections to other subject areas:* • *Art and Design – Visual communication, colour theory, composition, layout and digital creativity.* • *Photography – Camera positioning, lighting techniques, framing and image composition.* • *Media Studies – Animation, audience awareness and digital content creation.* • *Mathematics – Scale, coordinates, dimensions,*	*<u>Connections</u>* *We have identified these connections to other subject areas:* • *Mathematics – Logical reasoning, algorithms, variables, sequence, iteration and problem solving.* • *Design Technology – Using flowcharts and planning techniques to develop solutions and processes.* • *English – Following structured instructions, sequencing information*	*<u>Connections</u>* *We have identified these connections to other subject areas:* • Mathematics – Binary numbers, calculations, formulae, percentages, data handling and statistical analysis. • Science – Recording observations, presenting findings and analysing experimental data. • Geography – Interpreting data sets, creating charts and identifying trends..

• PSHE – Online safety, recognising scams, phishing and social engineering techniques. • Law – Introduction to the principles behind the Data Protection Act and responsible use of personal information. • Science – Collecting, analysing and evaluating data through investigative processes.	*transformations and geometric modelling.* • *Design Technology – Product design, modelling, prototyping and iterative development.* • *English – Evaluating products and communicating design decisions using appropriate terminology.*	*and using technical vocabulary accurately.* • *Modern Foreign Languages (MFL) – Understanding syntax, structure and translation between languages and coding systems.* • *Science – Modelling systems, predicting outcomes and testing solutions.*	• Music – Digital audio, sampling, sound representation and file formats. • Art and Photography – Digital images, pixels, resolution, colour depth and image manipulation.
<u>Learning at home</u> Consolidation of learning will take place at home by: • Completing online homework, and/or sheets which run concurrently with the lesson structure.	<u>*Learning at home*</u> *Consolidation of learning will take place at home by…* • Completing online homework, and/or sheets which run concurrently with the lesson structure.	<u>*Learning at home*</u> *Consolidation of learning will take place at home by…* • Completing online homework, and/or sheets which run concurrently with the lesson structure.	<u>*Learning at home*</u> *Consolidation of learning will take place at home by…* • Completing online homework, and/or sheets which run concurrently with the lesson structure.
<u>Sequencing</u> The Year 9 curriculum has been sequenced to consolidate and extend the programming, networking and digital media knowledge developed in Years 7 and 8, while introducing concepts that directly support progression into Key Stage 4 Information Technology. Learners begin the year by exploring cyber security and data science, developing an understanding of how data is collected, analysed, protected and interpreted in real-world contexts. This provides a strong foundation for later study of data handling, testing, cyber security and legislation. Learners then build their digital media skills through 3D design and animation, developing creativity alongside technical proficiency in the creation and evaluation of digital assets. The curriculum then progresses to flowcharts and advanced Python programming, allowing learners to strengthen their computational thinking and problem-solving skills. Introducing flowcharts provides learners with formal planning and design techniques that support algorithm development and establish a clear foundation for the design tools studied at Key Stage 4.			

Finally, learners study data representation and spreadsheet modelling. This develops an understanding of how images, audio and other forms of digital data are stored and processed by computer systems, while also introducing the collection, analysis and presentation of data through spreadsheets.

This sequencing ensures learners leave Key Stage 3 with secure foundations in cyber security, design, programming, data handling and digital media. It provides a clear bridge into the Key Stage 4 curriculum, particularly in the areas of design tools, data and testing, cyber security, digital communications and spreadsheet development.