PROUD TO BE PART OF
THE BISHOP FRASER TRUST
A CHURCH OF ENGLAND MULTI ACADEMY TRUST

Curriculum Intent

Computing

LET YOUR LIGHT SHINE
LUCEAT LUX VESTRA

BURY CE HIGH SCHOOL

Curriculum Vision

The vision for curriculum is:

- Students will **uphold strong Christian values**, always striving to **let their light shine**
- Students will be **responsible citizens** who **respect** everybody in school, the wider community and beyond
- Students will be **mentally** and **physically resilient**
- Students will experience **creativity and curiosity**
- Students will experience a **knowledge rich curriculum** that is broad and balanced, deepens over time and exposes them to a **wide variety of texts**

Departmental Vision

Our vision is to inspire curiosity, confidence and creativity with technology. We want every pupil to leave Bury Church of England High School able to understand, use and shape the digital world. We aim to develop pupils who think computationally and solve problems with resilience. We also want them to use emerging technologies such as artificial intelligence safely, ethically and with a critical eye. This prepares them for further study, the world of work, and life as responsible digital citizens.

Key Aspects of Our Vision:

1. ***Computational Thinking & Programming:*** *We develop logical thinking and problem-solving through decomposition, abstraction and algorithm design. Pupils begin with block-based programming and progress to text-based Python, equipping them with the skills to solve real-world problems.*
2. ***Understanding How Technology Works:*** *Pupils learn what happens inside the devices they use every day. They will study how data such as numbers, text, images and sound is represented in binary form, how computer systems and networks operate, and how to keep systems secure from cyber threats.*
3. ***Digital Skills:*** *Pupils build the essential IT skills needed in every workplace, including presenting information, handling data and producing digital media. They apply these skills creatively to design a variety of digital products for different audiences.*
4. ***Responsible Digital & AI Citizenship:*** *We teach safe, respectful and responsible use of technology, including protecting online identity and privacy and knowing how to report concerns. We also look at how AI tools work and where they make mistakes or show bias. Pupils will learn to use AI to support their thinking rather than letting it do the thinking for them.*
5. ***Innovation, Opportunity & Lifelong Learning:*** *We want every pupil to see a future for themselves in technology. Pupils can take part in a range of enrichment activities including entering national competitions such as the Bebras Computational Thinking Challenge and events celebrating computing across the academic year. Above all, we encourage a growth mindset so that pupils can adapt with confidence as technology keeps changing.*

At KS4, pupils can study GCSE Computer Science. This course builds directly on the programming, data representation and computer systems knowledge developed at Key Stage 3. All pupils at KS4 continue to develop their digital literacy, online safety and effective use of digital tools, preparing them for further study and future careers.

Curriculum Design

Key Stage 3 Computing

At Bury Church of England High School, our Computing curriculum is built on the three fundamental pillars of Computer Science, Information Technology and Digital Literacy. These pillars form the foundation of the National Curriculum for Computing. Our curriculum is designed so that pupils know and remember more over time, with knowledge in each pillar supporting and strengthening the others.

- ***Computer Science** is the core of the curriculum. It covers data representation, Boolean logic, algorithms, programming and computer systems, and provides the foundation for understanding every other area of computing.*
- ***Information Technology** is about how computers are used in society and in the workplace to create, collect and analyse information. Pupils will use a variety of online tools and create digital artefacts with attention to their audience, design and usability.*
- ***Digital Literacy** is the knowledge and ability to use technology confidently, competently and safely. It also covers online safety and privacy, evaluating the trustworthiness of information, and the safe, ethical and critical use of artificial intelligence.*

Key Stage 4 – GCSE Computer Science

Pupils are growing up in a world where technology is evolving rapidly, with new disciplines to explore and changes in the the way people work in every area, from medicine and fashion to engineering and economics.

GCSE Computer Science explores the principles of digital technology and a way of working called 'computational thinking', with programming at the core of the course. Pupils need to think logically, solve problems and be resilient when the going gets tough. It is also a very creative subject, and there is a real enjoyment to be had in getting something to work yourself, especially when programming. If pupils have enjoyed the programming they have completed in Key Stage 3, Computer Science could be an option to seriously consider.

The course builds directly on the programming, data representation and computer systems knowledge developed at Key Stage 3, and prepares pupils for further study and future careers.

Pupils follow the OCR GCSE Computer Science (J277) specification, assessed by two written examinations:

- ***Paper 1 – Computer Systems:*** *systems architecture, memory and storage, networks, network security, systems software, and the ethical, legal, cultural and environmental impacts of digital technology.*

- ***Paper 2 – Computational Thinking, Algorithms and Programming:*** *algorithms, programming fundamentals, producing robust programs, Boolean logic, and programming languages and IDEs.*

Within the course we aim for pupils to:

- *understand and apply the fundamental principles and concepts of computer science, including abstraction, decomposition, logic, algorithms and data representation*
- *analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs in Python*
- *think creatively, innovatively, analytically, logically and critically*
- *understand the components that make up digital systems, and how they communicate with one another and with other systems*
- *understand the impacts of digital technology to the individual and to wider society*
- *apply mathematical skills relevant to computer science*

Key Stage 4 – Digital Literacy

In Key Stage 4, all pupils have the opportunity to study aspects of Information Technology and Computer Science in sufficient depth to allow them to progress to higher levels of study or to a professional career. Alongside the GCSE Computer Science option, all pupils in Years 10 and 11 take part in Digital Literacy sessions, which continue to develop their digital literacy, online safety and effective use of digital tools, preparing them for further study and future careers. These sessions allow pupils to:

- *develop their capability, creativity and knowledge in computer science, digital media and information technology*
- *develop and apply their analytic, problem-solving, design and computational thinking skills*
- *understand how changes in technology affect safety, including new ways to protect their online privacy and identity, and how to report a range of concerns*

Year 7 Overview

Topic title	**Unit 7.1 – Introduction to Computing & E-Safety**
Declarative knowledge	• Know that social engineering is manipulating people into giving away private information, and that phishing and advance fee fraud are common examples • Know that a digital footprint is the permanent trail of data left by everything we do online, and that it can affect our future (positively and negatively) • Know the impact of fake news, and why false content spreads online • Know that malware is software designed to cause harm, and recognise common types (e.g. virus, trojan, ransomware, spyware) • Know the key ways to reduce the risks of malware e.g. anti-malware software, firewalls • Know what cyberbullying is, the effect it has on others, and the ways to report concerns • Know that a strong password uses three random words combined with numbers and symbols, and should never be shared or reused • Know ways to ensure we are upstanders when witnessing unkind online behaviour
Procedural knowledge	• Be able to access the school network, email system, Microsoft Teams and OneNote and be able to follow the rules of acceptable use. • Log on, complete and organise work within OneNote and Teams, save and access work in cloud storage i.e. OneDrive. • Identify the warning signs of a phishing email or scam message • Check whether a news story is trustworthy by checking its source, author and date, and comparing it with other reports • Respond to and report examples of unkind or unsafe online behaviour via the appropriate channels
Prior knowledge	**KS2:** use technology safely, respectfully and responsibly; recognise acceptable and unacceptable behaviour; identify a range of ways to report concerns about content and contact
Future knowledge	**Linked Units include:** Year 7 – Introduction to Artificial Intelligence (AI) Year 9 – Computer Networks & Cyber Security KS4 Digital Literacy GCSE – 1.4 Network Security; 1.6 Ethical, Legal, Cultural and Environmental Impacts
Key vocabulary	E-Safety, Social engineering, Phishing, Advanced Fee Fraud, Digital footprint, Passive Footprint, Active Footprint, Fake News, Source, Cyberbullying, Malware, Virus, Spyware, Ransomware, Trojan Horse, Cyberbullying, Bystander, Upstander

Topic title	**Unit 7.2 – Data Representation**
Declarative knowledge	• Know that computers use binary because their circuits (transistors) have two states: on (1) and off (0) • Know that a bit is a single binary digit, a nibble is 4 bits and a byte is 8 bits • Know that denary is base 10 and binary is base 2, and know the 8-bit place values (128, 64, 32, 16, 8, 4, 2, 1) • Know that a logic gate takes one or more binary inputs and produces one output, and know the rules for AND, OR and NOT • Know that a truth table shows the output of a logic gate for every possible combination of inputs
Procedural knowledge	• Convert numbers between 8-bit binary and denary • Complete truth tables for AND, OR and NOT gates • Complete truth tables for logic gate diagrams consisting of a number of gates • Draw basic logic gate diagrams to represent Boolean expressions
Prior knowledge	**KS2 Maths:** place value and number Year 7 – Introduction to Computing & E-Safety
Future knowledge	**Linked Units include:** Year 8 – Computer Systems Year 9 – Advanced Data Representation GCSE – 1.2 Memory and Storage (Data Storage); 2.4 Boolean Logic
Key vocabulary	Binary, Denary, Base, Bit, Nibble, Byte, Logic gate, Truth table, Boolean, AND, OR, NOT, Input, Output

Topic title	**Unit 7.3 – Intro to Computational Thinking & Programming (with Edublocks)**
Declarative knowledge	• Know that an algorithm is series of steps to solve a problem • Know that sequence means instructions are carried out in order, one after another • Know that iteration means repeating instructions, and that loops can repeat a set number of times (using FOR loops) • Know that selection uses a condition (True or False) to decide which instructions to run (IF statements) • Know that a variable is a named location in memory that stores a value which can change while the program runs • Know that data is often of two data types – numerical or text based • Know that input allows a user to enter data into the program, and output displays results/information back to the user • Know that Edublocks blocks represent real Python code • Know that decomposition is breaking a problem down into smaller, more manageable parts (a form of computational thinking) • Know that abstraction is removing unnecessary details to focus on what is important when problem solving (a form of computational thinking)
Procedural knowledge	• Build programs in Edublocks using sequence, iteration, selection, input and output. • Use correct comparison operators when designing IF statements (selection) • Predict the outcome of a program • Spot errors within a program • Identify sequence, selection, iteration and variables within a program • Decompose an everyday problem into smaller steps • Apply abstraction to identify the important information when problem solving
Prior knowledge	**KS2:** design, write and debug programs that accomplish specific goals; use sequence, selection and repetition in programs; use logical reasoning to explain how simple algorithms work
Future knowledge	**Linked Units include:** Year 7 – Python Programming Fundamentals Year 8 – Python Programming 101 GCSE – 2.1 Algorithms; 2.2 Programming Fundamentals
Key vocabulary	Programming, Algorithm, Debugging, Sequence, Selection, Iteration, Program, Loop, Condition, IF statement, Variable, Data Type, Input, Output, Decomposition, Abstraction

Topic title	**Unit 7.4 – Python Programming Fundamentals**
Declarative knowledge	• Know the arithmetic operators (+, –, *, /) and how to use them in Python • Know the comparison operators (==, !=, <, >, <=, >=) and how to use them in a Python IF statement • Know that variables are of a certain data type (integer, float or string) • Know that a syntax error breaks the rules of the language • Know that a logic error makes the program produce the wrong result/output • Know the standard flowchart symbols (terminator, process, input/output, decision) and what each represents • Know what sequence and selection look like within a Python program.
Procedural knowledge (Python code)	• Create variables with meaningful names and assign values of various data types to them. • Use input to store data in a variable • Use output to display appropriate information to the user • Perform calculations using arithmetic operators • Use comparison operators in IF statements to control the flow of a program • Find and fix syntax and logic errors in a program (debugging) • Draw and trace flowcharts that represent simple programs
Prior knowledge	Year 7 – Intro to Computational Thinking & Programming
Future knowledge	**Linked Units include:** Year 8 – Python Programming 101 Year 9 – Advanced Python Programming GCSE – 2.2 Programming Fundamentals; 2.3 Producing Robust Programs
Key vocabulary	Debugging, Sequence, Selection, Program, Condition, IF statement, Variable, Data Type, Integer, Float, String, Input, Output, Operator, Flowchart, Syntax error, Logic Error

Topic title	**Unit 7.5 – Introduction to Artificial Intelligence (AI)**
Declarative knowledge	• Know that artificial intelligence (AI) describes computer systems that perform tasks which normally need human intelligence • Know that machine learning is a type of AI where a model learns patterns from training data rather than following hand-written rules • Know the basics behind 3 types of machine learning – supervised, unsupervised and reinforcement learning. • Know that the quality of a model depends on its training data, and that bad data can cause bias • Know that generative AI creates new text, images or sound, and can produce convincing but incorrect results (hallucinations) • Know the ethical issues linked to AI, including plagiarism, deepfakes, privacy and the impact on jobs • Know that relying on AI to do your thinking (cognitive offloading) can stop you learning • Know that AI tools can be wrong, may have age restrictions, and should never be given personal information
Procedural knowledge	• Train a simple machine learning model to classify data, then test and improve its accuracy • Show how changing the training data changes the results • Spot mistakes in AI-generated content • Discuss an ethical AI scenario, giving reasons for and against • Decide whether a task is a sensible use of AI or should be done independently
Prior knowledge	Year 7 – Introduction to Computing & E-Safety Year 7 – Computational Thinking with Edublocks
Future knowledge	**Linked Units include:** Year 8 – Data Science Project KS4 Digital Literacy GCSE – 1.6 Ethical, Legal, Cultural and Environmental Impacts
Key vocabulary	Artificial intelligence, Machine learning, Supervised, Unsupervised, Reinforcement, Training data, Bias, Generative AI, Hallucination, Model, Classification, Cognitive offloading

Topic title	**Unit 7.6 – Creative Media Project: Interactive Information Point**
Declarative knowledge	• Know that every digital product is designed for a specific audience and purpose • Know the principles of good design: consistency, colour contrast, readable sans-serif fonts etc. • Know that use of transitions and custom animations should be consistent and suitable for purpose and audience • Know that hyperlinks allow non-linear navigation, like an information kiosk • Know the basic image editing tools (crop, resize, rotate, adjust, remove background) and when to use them • Know that images found online are protected by copyright, and that Creative Commons licences allow some reuse with credit • Know that copying information from sources online without credit is plagiarism
Procedural knowledge	• Build an interactive PowerPoint information point with interactive hyperlinks • Select, credit and edit suitable images for a given audience • Test the product against success criteria
Prior knowledge	Year 7 – Introduction to Computing & E-Safety
Future knowledge	**Linked Units include:** Year 8 – Web Development 101 KS4 Digital Literacy
Key vocabulary	Audience, Purpose, Hyperlink, Navigation, Copyright, Creative Commons, plagiarism, consistency, relevancy, colour contrast, sans-serif, transition, custom animation

Year 8 Overview

Topic title	**Unit 8.1 – Python Programming 101**
Declarative knowledge	• Know that Python is a high-level, text-based programming language • Know that the print() function produces output • Know that the input() function allows user input (and always returns a string) • Know that a variable is a named memory location to store a value, that can change during the execution of the program • Know the Python variable naming constraints • Know the main data types: integer, float (real), string and Boolean • Know that casting converts data from one type to another (int(), float(), str()) • Know that concatenation joins strings together • Know the basic arithmetic operators (+, -, *, /) • Know that selection uses if, elif and else to choose which code runs based on a condition, and that indentation shows which code belongs to each branch • Know how to use comparison and Boolean operators (and, or, not) to construct a condition
Procedural knowledge	• Write programs that take input, process it and produce output • Choose and use appropriate data types and variables, casting input where needed • Use concatenation to build bespoke output messages • Write calculations using basic arithmetic operators • Use if, elif and else statements with comparison and Boolean operators (and, or, not) • Read error messages carefully and debug programs
Prior knowledge	Year 7 – Intro to Computational Thinking & Programming (with Edublocks) Year 7 – Python Programming Fundamentals
Future knowledge	**Linked Units include:** Year 9 – Advanced Python Programming & Key Algorithms GCSE – 2.2 Programming Fundamentals; 2.5 Programming Languages and IDEs
Key vocabulary	Input, Output, Variable, Data type, Integer, Float, String, Boolean, Casting, Concatenation, Selection, IF statement, Syntax, Indentation, Comparison Operators, Boolean Operators

Topic title	**Unit 8.2 – Web Development 101**
Declarative knowledge	• Know that HTML (HyperText Markup Language) describes the structure and content of a web page • Know that CSS (Cascading Style Sheets) controls the appearance and formatting of a web page • Know that a web browser requests pages from a web server and renders the HTML and CSS it receives • Know that HTML is made of elements written with tags • Know the structure of an HTML document (doctype, html, head, body) • Know that a CSS rule is made of a selector, properties and values • Know that websites must meet the needs of their purpose and audience
Procedural knowledge	• Write a web page by hand using headings, paragraphs, lists, images and hyperlinks • Style a webpage consistently using CSS • Test links and layout, and debug errors with HTML and CSS code
Prior knowledge	Year 7 – Creative Media Project (audience, navigation, hyperlinks) Year 8 – Python Programming 101 (writing and debugging text-based code)
Future knowledge	**Linked Units include:** Year 9 – Computer Networks & Cyber Security (the internet and the World Wide Web) GCSE – 1.3 Networks, Connections and Protocols
Key vocabulary	HTML, CSS, Tag, Element, Hyperlink, Web browser, Stylesheet, Web server, headings, paragraphs, ordered list, unordered list

Topic title	**Unit 8.3 – Computer Systems**
Declarative knowledge	• Know that a computer system is made of hardware (physical components) and software (programs) • Know the difference between input, output and storage devices, with examples of each • Know that the CPU (Central Processing Unit) processes instructions by repeatedly fetching, decoding and executing them • Know the difference between RAM (volatile, temporary) and ROM (non-volatile, stores start-up instructions) • Know that secondary storage keeps data permanently, and that the main types are magnetic, optical and solid state
Procedural knowledge	• Classify devices as input, output or storage • Identify the role of the main components inside a computer and how they communicate with each other • Explain the CPU fetch-decode-execute cycle in simple terms • Choose suitable storage for a given scenario
Prior knowledge	Year 7 – Intro to Computational Thinking & Programming (input and output)
Future knowledge	**Linked Units include:** Year 9 – Advanced Computer Systems GCSE – 1.1 Systems Architecture; 1.2 Memory and Storage; 1.5 Systems Software
Key vocabulary	Hardware, Software, CPU, RAM, ROM, Secondary storage, Input device, Output device, Fetch-decode-execute, Volatile, Solid state, Magnetic, Optical

Topic title	**Unit 8.4 – Data Science**
Declarative knowledge	• Know the difference between data (raw facts) and information (data with meaning) • Know that a dataset is a collection of data organised in rows and columns • Know that every spreadsheet cell has a unique reference, and that formula results update automatically when data changes • Know the common functions SUM, AVERAGE, MAX, MIN, COUNT and COUNTIF • Know that charts and conditional formatting help to identify patterns, trends, correlation and outliers (data visualisation) • Know that AI tools can help analyse data, but we must be mindful of data confidentiality and outputs must always be checked for accuracy and bias
Procedural knowledge	• Use tables, formulas, functions, sorting and filtering to analyse data • Use data visualisation to create suitable charts for given datasets • Apply conditional formatting
Prior knowledge	Year 7 – Introduction to Artificial Intelligence (training data and bias) Year 8 – Python Programming 101 (data types, operators)
Future knowledge	**Linked Units include:** GCSE – 2.2 Programming Fundamentals (SQL, records) KS4 Digital Literacy Further study and employment requiring data skills
Key vocabulary	Data, Dataset, Information, Cell, Formula, Chart, Sort, Conditional formatting, Trend, Correlation, Outlier, Data Visualisation

Year 9 Overview

Topic title	**Unit 9.1 – Advanced Python Programming & Key Algorithms**
Declarative knowledge	• Recall all knowledge from 8.1 Python Programming 101 • Know that count-controlled iteration (for loops) repeats a fixed number of times and condition-controlled iteration (while loops) repeats until a condition is no longer True • Know that while loops can be used for validation checks on inputs (e.g. range, type, presence and length checks) • Know that a list (array) stores multiple items under one name, and that each item is accessed by its index (position), starting at 0 • Know that a subprogram (function or procedure) is a named block of code that can be reused, and that parameters pass data into it • Know that a function returns a value and a procedure does not • Know how a linear search checks each item in turn, and how an algorithm finds the maximum or minimum value
Procedural knowledge	• Write programs using for and while loops, including nested selection inside loops • Create, access, update and loop through lists • Use functions with parameters and return values • Write validation routines for user input • Recognise linear search and a find maximum/minimum algorithm • Complete trace tables to follow the execution of a program
Prior knowledge	Year 8 – Python Programming 101 Year 7 – Programming Fundamentals Year 7 – Intro to Computational Thinking & Programming
Future knowledge	**Linked Units include:** GCSE – 2.1 Algorithms; 2.2 Programming Fundamentals; 2.3 Producing Robust Programs
Key vocabulary	Iteration, For loop, While loop, List, Index, Function, Parameter, Validation, Linear search, Procedure, Return value, Trace table

Topic title	**Unit 9.2 – Advanced Data Representation**
Declarative knowledge	• Recall knowledge from Year 7 – Data Representation • Know the units of data storage (bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte, petabyte), and that each unit is 1000 times the previous one • Know that hexadecimal is base 16 and is used because it is a shorter, more readable way to represent binary (for humans) • Know that binary addition can cause overflow when a result needs more bits than are available • Know that a character set is the list of characters recognised by a computer • Know that a bitmap image is made of pixels, and that resolution and colour depth affect quality and file size • Know that compression reduces file size, and the difference between lossy and lossless compression
Procedural knowledge	• Convert between binary, denary and hexadecimal • Perform binary addition and identify overflow • Calculate the file size of an image file • Convert units of data storage • Discuss the effects of lossless and lossy compression on an image
Prior knowledge	Year 7 – Data Representation
Future knowledge	**Linked Units include:** GCSE – 1.2 Memory and Storage (Data Storage); 2.4 Boolean Logic
Key vocabulary	Hexadecimal, Binary Addition, Overflow, Character Set, Pixel, Resolution, Colour/Bit depth, Compression, Lossy, Lossless

Topic title	**Unit 9.3 – Advanced Computer Systems**
Declarative knowledge	• Recall knowledge from Year 8 – Computer Systems • Know the performance factors of the CPU (clock speed, cache, cores) and how they affect the capability of the CPU • Know the stages of the fetch-decode-execute cycle • Know that virtual memory uses secondary storage when RAM is full, and why this slows the computer down • Know the main categories of software, their purpose and examples of each – application software and systems software • Know the main functions of an operating system (user interface, memory management, peripheral management, user management, file management) • Know the main types of user interface – command line, GUI (WIMP), menu-driven, natural language
Procedural knowledge	• Compare computer specifications and justify which will perform better • Explain when and why virtual memory is used • Explain the difference between application and systems software • Identify the main functions of an OS • Identify the different types of User Interface that would be suitable for a given computer system
Prior knowledge	Year 8 – Computer Systems
Future knowledge	**Linked Units include:** GCSE – 1.1 Systems Architecture; 1.2 Memory and Storage; 1.5 Systems Software
Key vocabulary	Cache, Cores, Clock speed, Virtual memory, Application, Systems Software, Utility software, Operating System, Fetch-decode-execute, User Interface, GUI, WIMP

Topic title	**Unit 9.4 – Computer Networks & Cyber Security**
Declarative knowledge	• Know that a network is two or more connected devices • Know the difference between a LAN and a WAN • Know the purpose of network hardware: router, wireless access point and network interface card • Know the difference between the internet (the network of interconnected networks) and the World Wide Web (the pages accessed over it) • Know that data is sent across networks in packets, and that devices are identified on a network by IP addresses • Know the advantages and disadvantages of wired and wireless connections • Know common cyber threats: malware, social engineering, brute-force attacks, hacking and denial of service (DoS) • Know how to prevent attacks: firewalls, anti-malware, strong passwords and two-factor authentication, encryption
Procedural knowledge	• Identify categories of network e.g. LAN vs WAN, Wired Vs Wireless • Explain the purpose of key network hardware • Identify threats in a case study and recommend suitable prevention methods
Prior knowledge	**KS2:** 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 Year 7 – Introduction to Computing & E-Safety Year 8 – Web Development 101
Future knowledge	**Linked Units include:** GCSE – 1.3 Networks, Connections and Protocols; 1.4 Network Security; 1.6 Ethical, Legal, Cultural and Environmental Impacts KS4 Digital Literacy
Key vocabulary	Network, LAN, WAN, Router, WAP, IP address, Internet, World Wide Web, Wired, Wireless, Malware, Firewall, Anti-malware, 2FA, Encryption, Packet, Brute-force attack, Denial of service

Year 10 Computer Science GCSE

Topic title	**J277 1.1 – Systems Architecture**
Declarative knowledge	• Know the purpose of the CPU and the fetch-execute cycle • Know the common CPU components and their function: ALU, CU, cache and registers • Know the Von Neumann architecture and the purpose of the MAR, MDR, program counter and accumulator • Know how clock speed, cache size and number of cores affect CPU performance • Know the purpose and characteristics of embedded systems, with examples
Procedural knowledge	• Explain what happens in each stage of the fetch-execute cycle, including the role of each register • Explain how changing a performance factor affects the CPU • Answer OCR exam-style questions applying knowledge to unfamiliar scenarios
Prior knowledge	Year 8 – Computer Systems Year 9 – Advanced Computer Systems
Future knowledge	A Level Computer Science or equivalent qualification
Key vocabulary	CPU, Fetch-execute cycle, ALU, Control unit, Cache, Register, Von Neumann architecture, MAR, MDR, Program counter, Accumulator, Clock speed, Core, Embedded system

Topic title	**J277 1.2 – Memory and Storage (Primary and Secondary Storage)**
Declarative knowledge	• Know the need for primary storage, and the difference between RAM and ROM • Know the purpose of virtual memory • Know the need for secondary storage, and the common types: optical, magnetic and solid state • Know the characteristics of storage: capacity, speed, portability, durability, reliability and cost • Know the units of data storage, from bit to petabyte, and why data must be stored in binary
Procedural knowledge	• Recommend suitable storage for a given scenario and justify the choice using the characteristics • Calculate required storage capacity for a set of files • Convert between units of data storage
Prior knowledge	Year 8 – Computer Systems Year 9 – Advanced Computer Systems Year 9 – Advanced Data Representation
Future knowledge	A Level Computer Science or equivalent qualification
Key vocabulary	Primary storage, RAM, ROM, Virtual memory, Secondary storage, Volatile, Non-volatile, Optical, Magnetic, Solid state, Capacity, Durability, Portability

Topic title	**J277 1.2 – Memory and Storage (Data Storage)**
Declarative knowledge	• Know how binary is used to represent numbers, and how to convert between binary, denary and hexadecimal • Know how to add two binary integers (up to 8 bits) and what an overflow error is • Know the effect of binary shifts and how to apply them • Know how characters are represented using a character set, and the differences between ASCII and Unicode • Know how images are represented using pixels, colour depth and resolution, and the role of metadata • Know how sound is sampled and stored, and how sample rate, duration and bit depth affect quality and file size • Know the need for compression and the difference between lossy and lossless compression
Procedural knowledge	• Convert between binary, denary and hexadecimal (up to 8 bits / 2 hex digits) • Add two binary integers (up to 8 bits) and explain overflow errors • Apply left and right binary shifts and explain the effect it has had • Calculate the file size of images and sound files • Choose an appropriate type of compression for a scenario and justify the choice
Prior knowledge	Year 7 – Data Representation Year 9 – Advanced Data Representation
Future knowledge	A Level Computer Science or equivalent qualification
Key vocabulary	Binary, Denary, Hexadecimal, Overflow, Binary shift, Character set, ASCII, Unicode, Colour depth, Resolution, Sample rate, Bit depth, Compression, Metadata, Lossy, Lossless

Topic title	**J277 2.4 – Boolean Logic**
Declarative knowledge	• Know the logic gate symbols and truth tables for AND, OR and NOT • Know how logic gates can be combined into simple logic diagrams • Know how Boolean operators are applied in truth tables and expressions
Procedural knowledge	• Draw and interpret simple logic diagrams using AND, OR and NOT gates • Complete truth tables for logic diagrams and Boolean expressions • Write the Boolean expression for a logic diagram, and draw the diagram for an expression
Prior knowledge	Year 7 – Data Representation (logic gates) Year 9 – Advanced Data Representation
Future knowledge	A Level Computer Science or equivalent qualification
Key vocabulary	Logic gate, AND, OR, NOT, Truth table, Boolean expression, Logic diagram, Input, Output

Topic title	**J277 1.3 – Networks, Connections and Protocols**
Declarative knowledge	• Know the types of network (LAN and WAN) and the factors that affect network performance • Know the different roles of computers in client-server and peer-to-peer networks • Know the hardware needed to connect stand-alone computers into a LAN: wireless access point, router, switch, NIC and transmission media • Know that the internet is a worldwide collection of computer networks, and the role of DNS, hosting and the cloud • Know the star and mesh topologies • Know about wired and wireless connections (Ethernet, Wi-Fi, Bluetooth) and the role of encryption • Know about IP addressing and MAC addressing, • Know the importance of standards in networking • Know the common protocols (TCP/IP, HTTP, HTTPS, FTP, POP, IMAP, SMTP) and the concept of layers
Procedural knowledge	• Draw and compare star and mesh topologies, explaining their advantages and disadvantages • Explain how DNS is used to find a web page • Select the appropriate protocol for a given task • Explain the benefits of using layers
Prior knowledge	Year 8 – Web Development 101 Year 9 – Computer Networks & Cyber Security
Future knowledge	A Level Computer Science or equivalent qualification
Key vocabulary	LAN, WAN, Client-server, Peer-to-peer, Router, Switch, DNS, Topology, Protocol, IP address, MAC address, Wireless access point, NIC, Bandwidth, Cloud, TCP/IP, HTTP, HTTPS, Layers

Topic title	**J277 2.1 – Algorithms**
Declarative knowledge	• Know the principles of computational thinking: abstraction, decomposition and algorithmic thinking • Know how to identify the inputs, processes and outputs for a problem, and how to use structure diagrams • Know how algorithms are represented using pseudocode, flowcharts and OCR Exam Reference Language • Know how the standard searching algorithms work: binary search and linear search • Know how the standard sorting algorithms work: bubble sort, merge sort and insertion sort
Procedural knowledge	• Create, interpret, correct, complete and refine algorithms using flowcharts and code • Use trace tables to follow an algorithm and identify errors • Carry out each searching and sorting algorithm on a given data set, explaining the steps taken • Recognise the different standard algorithms and their key features
Prior knowledge	Year 7 – Intro to Computational Thinking & Programming Year 8 – Python Programming 101 Year 9 – Advanced Python Programming & Key Algorithms
Future knowledge	A Level Computer Science or equivalent qualification
Key vocabulary	Abstraction, Decomposition, Algorithmic thinking, Pseudocode, Flowchart, Trace table, Linear search, Binary search, Bubble sort, Merge sort, Insertion sort, Structure diagram

Topic title	**J277 2.2 – Programming Fundamentals (Part 1: Core Constructs)**
Declarative knowledge	• Know the use of variables, constants, operators, inputs, outputs and assignments • Know the three basic programming constructs: sequence, selection and iteration (count-controlled and condition-controlled) • Know the common arithmetic operators, including MOD (%), DIV (//) and exponentiation (**) • Know the comparison and Boolean operators (AND, OR, NOT) • Know the data types integer, float/real, Boolean, character and string, and the need for casting • Know that one-dimensional and two-dimensional arrays store multiple values of the same type • Know the OCR Exam Reference Language equivalents of the Python constructs used • Know that records store related data of different types, and how SQL is used to search for data in a database
Procedural knowledge	• Write Python programs that use sequence, selection and iteration, including nested structures • Use the correct data type for a given purpose and cast between types • Use basic string manipulation, including concatenation and slicing • Write programs that use 1D and 2D arrays • Write SQL queries using SELECT, FROM and WHERE
Prior knowledge	Year 8 – Python Programming 101 Year 9 – Advanced Python Programming & Key Algorithms
Future knowledge	**Linked Units include:** Year 11 – J277 2.2 Programming Fundamentals (Part 2) A Level Computer Science or equivalent qualification
Key vocabulary	Variable, Constant, Assignment, Sequence, Selection, Iteration, Data type, Casting, MOD, DIV, Integer, Real, Boolean, Character, String, Concatenation, Array, 2D Array, SQL, SELECT, FROM, WHERE

Year 11 Computer Science GCSE

Topic title	**J277 1.4 – Network Security**
Declarative knowledge	• Know the forms of attack: malware, social engineering (including phishing), brute-force attacks, denial of service, data interception and theft, and SQL injection • Know the common prevention methods: penetration testing, anti-malware software, firewalls, user access levels, passwords, encryption and physical security
Procedural knowledge	• Explain how each threat works and the problems it causes • Identify and justify suitable prevention methods for a given threat or scenario
Prior knowledge	Year 7 – Introduction to Computing & E-Safety Year 9 – Computer Networks & Cyber Security
Future knowledge	A Level Computer Science or equivalent qualification Careers in cyber security
Key vocabulary	Malware, Social engineering, Phishing, Brute-force attack, Denial of service, SQL injection, Firewall, Encryption, Penetration testing, Data interception, User access levels, Anti-malware, Physical security

Topic title	**J277 1.5 – Systems Software**
Declarative knowledge	• Know the purpose and functions of operating systems: user interface, memory management and multitasking, peripheral management and drivers, user management, and file management • Know the purpose and functionality of utility software: encryption software, defragmentation and data compression
Procedural knowledge	• Explain how the operating system manages each resource • Explain why each utility may be required for a given scenario • Explain how defragmentation improves performance on a magnetic hard disk
Prior knowledge	Year 8 – Computer Systems Year 9 – Advanced Computer Systems
Future knowledge	A Level Computer Science or equivalent qualification
Key vocabulary	Operating system, Utility software, Memory management, Multitasking, Device driver, Defragmentation, User interface, Peripheral, File management, Encryption, Compression

Topic title	**J277 1.6 – Ethical, Legal, Cultural and Environmental Impacts of Digital Technology**
Declarative knowledge	• Know the impacts of digital technology on wider society, including ethical, legal, cultural, environmental and privacy issues • Know the purpose of the Data Protection Act 2018, the Computer Misuse Act 1990 and the Copyright, Designs and Patents Act 1988 • Know the difference between open source and proprietary software licences • Know the impact of emerging technologies, including artificial intelligence, on individuals and society
Procedural knowledge	• Discuss and evaluate the impacts of technology in extended-response questions, considering different stakeholders • Apply the correct legislation to a given scenario
Prior knowledge	Year 7 – Introduction to Artificial Intelligence Year 9 – Computer Networks & Cyber Security (legislation)
Future knowledge	A Level Computer Science or equivalent qualification Informed and responsible citizenship
Key vocabulary	Ethical, Legal, Cultural, Environmental, Privacy, Data Protection Act, Computer Misuse Act, Copyright, Designs and Patents Act, Open source, Proprietary, Stakeholder, E-waste, Digital divide

Topic title	**J277 2.2 – Programming Fundamentals (Part 2: Data Structures, Files and Subprograms)**
Declarative knowledge	• Know the purpose of file handling: opening, reading, writing and closing files • Know how to use subprograms (functions and procedures), including parameters and return values • Know the difference between local and global variables • Know how to use random number generation
Procedural knowledge	• Write programs that read from and write to text files for a given scenario • Use string manipulation and random number generation to solve problems • Structure programs using functions and procedures with parameters
Prior knowledge	Year 9 – Advanced Python Programming & Key Algorithms (lists, functions) Year 8 – Data Science Project (data and records) Year 10 – J277 2.2 Programming Fundamentals (Part 1)
Future knowledge	A Level Computer Science or equivalent qualification
Key vocabulary	File handling, Subprogram, Function, Procedure, Parameter, Local variable, Global variable, Random

Topic title	**J277 2.3 – Producing Robust Programs**
Declarative knowledge	• Know the principles of defensive design: anticipating misuse, authentication and input validation • Know how to make programs maintainable: subprograms, naming conventions, indentation and commenting • Know the purpose of testing, and the difference between iterative and final/terminal testing • Know the difference between syntax and logic errors • Know the types of test data: normal, boundary, invalid and erroneous
Procedural knowledge	• Write input validation and simple authentication routines • Select and justify suitable test data for a given program, and create test data for a test plan • Identify, categorise and correct errors in programs • Refine algorithms and programs to make them more robust
Prior knowledge	Year 9 – Advanced Python Programming & Key Algorithms (validation, test data) Year 10 – J277 2.2 Programming Fundamentals (Part 1) Year 11 – J277 2.2 Programming Fundamentals (Part 2)
Future knowledge	A Level Computer Science or equivalent qualification
Key vocabulary	Defensive design, Validation, Authentication, Maintainability, Syntax error, Logic error, Test data, Normal, Boundary, Invalid, Erroneous, Iterative testing, Final testing, Test plan

Topic title	**J277 2.5 – Programming Languages and Integrated Development Environments**
Declarative knowledge	• Know the characteristics and purpose of high-level and low-level languages • Know the purpose of translators, and the characteristics of compilers and interpreters • Know the common tools and facilities available in an IDE: editors, error diagnostics, run-time environment and translators
Procedural knowledge	• Compare high-level and low-level languages, and compilers and interpreters, for a given situation • Explain how IDE tools help programmers to write and debug code
Prior knowledge	Year 9 – Advanced Python Programming & Key Algorithms Year 10 – J277 2.2 Programming Fundamentals (Part 1) Year 11 – J277 2.2 Programming Fundamentals (Part 2)
Future knowledge	A Level Computer Science or equivalent qualification
Key vocabulary	High-level language, Low-level language, Machine code, Translator, Compiler, Interpreter, IDE, Assembly language, Error diagnostics, Run-time environment