

Programme of study for Year 8 Computer Science and IT

Autumn (1 st term) Topic	Autumn (2 nd term) Topic	Spring (1 st term) Topic	Spring (2 nd term) Topic	Summer (1 st term) Topic	Summer (2 nd term) Topic
Computer Networks (6) Exploring how computers communicate – from the Internet and World Wide Web to addressing, key protocols and the security risks that come with being connected.	HTML and Web Development (6) – Part 1 Now that pupils understand how the Internet and WWW work, they learn to build for it – starting with the building blocks of a webpage: HTML and CSS.	HTML and Web Development (6) – Part 2 Pupils apply their HTML/CSS skills to plan, design and build a complete webpage, then evaluate it against design principles.	Data Representation (6) Looking “under the hood” at how computers store the text and images pupils have just been publishing on their webpages – as binary.	Introduction to Python Programming (6) Moving from Year 7’s block-based Scratch into proper text-based code, learning core programming constructs in Python, with a single taster lesson at the end trying Python out on a real micro:bit.	Data Protection, Cybersecurity and Encryption (6) Understanding the laws that protect personal data, the threats that put it at risk, and the encryption techniques – now codeable in Python – used to keep it secure.
Skills: Explaining the difference between the Internet and the World Wide Web. Understanding domains and IP addresses. Describing how email and VoIP transmit data across a network. Identifying common network protocols. Recognising network security threats and basic countermeasures.	Skills: Writing basic HTML to structure a webpage. Using CSS to style text, colour and layout. Understanding copyright when using online text and images.	Skills: Planning a webpage for a given audience and purpose. Combining HTML and CSS to build a multi-element webpage. Evaluating a webpage against design principles. Troubleshooting and debugging HTML/CSS code.	Skills: Converting between binary and denary. Encoding and decoding text using ASCII. Identifying how images are represented as binary. Understanding units of data storage.	Skills: Writing and running simple Python programs. Using sequence, selection (if/elif/else) and iteration (for/while loops). Storing and using data with variables. Trying out Python on a real micro:bit.	Skills: Understanding the Data Protection Act and Computer Misuse Act. Recognising ethical hacking and its purpose. Understanding encryption principles. Encoding/decoding messages using the Caesar Cipher, written as a Python program.
Key Learning Outcomes: Distinguish between the Internet (the global network of networks) and the World Wide Web (the pages and content accessed via it). Explain how a domain name is translated into an IP address using DNS. Describe how email and Voice over IP (VoIP) send data between devices on a network. Identify the role of key protocols	Key Learning Outcomes: Use core HTML tags to structure a simple webpage (headings, paragraphs, links, images, lists). Apply CSS to format text, colour and page layout. Explain what	Key Learning Outcomes: Plan a webpage for a specified audience and purpose. Build a webpage combining text, images, links and styled layout. Evaluate their own	Key Learning Outcomes: Convert binary numbers to denary and vice versa. Encode and decode simple text using ASCII character codes. Explain how a bitmap image is represented as binary (pixels and colour depth). Compare units of data storage (bit, byte, KB, MB, GB) and explain why they matter.	Key Learning Outcomes: Write simple Python programs using correct syntax. Use selection to make decisions in code (if/elif/else). Use iteration (for and while loops) to repeat actions. Store and update data using variables. In a single taster lesson, run a simple Python program on	Key Learning Outcomes: Explain the purpose of the Data Protection Act and the Computer Misuse Act. Understand the difference between ethical and unethical hacking. Explain why encryption is used to keep data secure. Encrypt and decrypt a message using the Caesar Cipher. Write a Python program that

Autumn (1 st term) Topic	Autumn (2 nd term) Topic	Spring (1 st term) Topic	Spring (2 nd term) Topic	Summer (1 st term) Topic	Summer (2 nd term) Topic
such as HTTP/HTTPS, TCP/IP and FTP. Explain common network security risks (e.g. malware, unauthorised access) and basic protective measures (e.g. firewalls, strong passwords, encryption).	copyright is and why it matters when reusing text, images or other content found online.	and others' webpages against design principles (e.g. readability, navigation, consistency). Identify and fix common HTML/CSS errors.		a real micro:bit to see code controlling physical hardware.	applies the Caesar Cipher to encode and decode text.
Half Term 1 Evidence to cover: Understanding of how data travels across the Internet (Internet vs WWW, domains and IPs) and a network protocols and security knowledge check.	Half Term 2 Evidence to cover: A simple webpage built using HTML and CSS, demonstrating correct use of tags and basic styling.	Half Term 3 Evidence to cover: A completed webpage project, evaluated against agreed design criteria.	Half Term 4 Evidence to cover: Binary–denary conversion exercises, an ASCII encoding/decoding task, and an explanation of how an image is stored as binary.	Half Term 5 Evidence to cover: A series of short Python programs demonstrating sequence, selection, iteration and variables, plus a short micro:bit taster activity.	Half Term 6 Evidence to cover: Understanding of data protection law and ethical hacking; a working Caesar Cipher program written in Python; end-of-year review.
Rationale for sequence: The year begins by building pupils' understanding of how computers actually connect and communicate, giving them the infrastructure knowledge that underpins everything else they do online this year – including the website they go on to design and build in the very next unit.	Rationale for sequence: Having just learned how the WWW delivers content from server to browser, pupils now create that content themselves – HTML and CSS are introduced first as the essential building blocks before any design work begins.	Rationale for sequence: This half-term builds directly on the HTML/CSS foundations from Autumn 2, moving pupils from isolated tags and styles to planning and building a complete, purposeful webpage – the natural next step before they turn to a new strand of “under the hood” knowledge in Spring 2.	Rationale for sequence: Having just published text and images on their own webpage, pupils now examine how that same content is actually stored by a computer – connecting the visible (a styled webpage) with the invisible (the binary beneath it) before this number-based thinking feeds into the formal logic of Python in the summer term.	Rationale for sequence: Year 7 ended by bridging from Scratch to a first look at text-based code; Year 8 picks up that thread immediately with a proper, dedicated introduction to Python – protecting the time needed to build real fluency, rather than spreading it thinly across a hardware project too. A single taster lesson with a micro:bit at the end lets pupils see their own code controlling physical hardware, without diverting teaching time away from the core language skills.	Rationale for sequence: The year closes by bringing two threads together – the legal/ethical side of keeping data safe, and the Python skills just learned in Summer 1 – with pupils writing their own encryption program. This also sets up Year 9, where the human side of security (social engineering) and further Python are explored in greater depth.
Home – Learning: Research and explain, step by step, what happens when you type a	Home – Learning: Use the W3Schools HTML tutorial to try	Home – Learning: Find and evaluate a real website, noting	Home – Learning: Practise binary–denary conversion using a free online game.	Home – Learning: Complete two beginner exercises on an online	Home – Learning:

Autumn (1 st term) Topic	Autumn (2 nd term) Topic	Spring (1 st term) Topic	Spring (2 nd term) Topic	Summer (1 st term) Topic	Summer (2 nd term) Topic
website address into a browser and press enter.	out three new tags not covered in class.	one design choice you would copy and one you would change.		Python platform of your choice.	Research one real-world data breach and summarise what went wrong.
Reading / High Quality Text: Computer Science: Networks – GCSE Bitesize https://www.bbc.co.uk/bitesize/guides/zr3yb82/revision/6 Literacy: Technical terms related to networks and protocols (domain, IP address, protocol, bandwidth, client-server).	Reading / High Quality Text: HTML Tutorial – W3Schools https://www.w3schools.com/html/ Literacy: Technical terms related to HTML and web development (tag, attribute, element, stylesheet).	Reading / High Quality Text: CSS Tutorial – W3Schools https://www.w3schools.com/css/ Literacy: Technical terms related to web design and evaluation (layout, navigation, usability, accessibility).	Reading / High Quality Text: Data Representation – BBC Bitesize KS3 Computer Science https://www.bbc.co.uk/bitesize/guides/z26redm/revision/1 Literacy: Technical terms related to data representation (bit, byte, binary, denary, ASCII, bitmap).	Reading / High Quality Text: Python Tutorial – W3Schools https://www.w3schools.com/python/default.asp First lessons with Python and the micro:bit https://microbit.org/teach/lessons/first-lessons-with-python-and-the-microbit/ Literacy: Technical terms related to Python and physical computing (variable, loop, function, sensor, syntax).	Reading / High Quality Text: Ethical Hacking – BBC Bitesize https://www.bbc.co.uk/bitesize/guides/zbgg4qt/revision/8 Computer Misuse Act 1990 – legislation.gov.uk https://www.legislation.gov.uk/ukpga/1990/18/contents History of the Caesar Cipher https://kids.kiddle.co/Caesar_cipher Literacy: Technical terms related to data protection and encryption (encryption, decryption, cipher, ethical hacking, legislation).
Numeracy: IP address structure (denary octets and their binary equivalents); comparing data transmission speeds (Mbps/Gbps).	Numeracy: Pixel measurements, colour codes (hex/RGB values), and layout proportions.	Numeracy: Page-loading times and file sizes; proportion and alignment in page layout.	Numeracy: Binary–denary conversion; storage units (bytes, KB, MB, GB) and their relative scale.	Numeracy: Using variables to count and calculate (e.g. scores, timers); logical/Boolean conditions (and, or, not) in selection statements.	Numeracy: Letter-shift calculations in the Caesar Cipher (modular arithmetic); statistics on cybersecurity incidents.
Enrichment / opportunities to develop cultural capital (including careers, WRL and SMSC): Participate in coding challenges online (e.g. CyberFirst, Bebras). Explore careers such as Network Engineer, Web Developer, Python Developer and Cybersecurity Analyst. Attend tech-related conferences, workshops or virtual careers talks.					