

Curriculum Map Year 10 GCSE COMPUTER SCIENCE

Topic Name	Term	Skills developed with link to NC Subject content	Reflection on previous link in the curriculum	Progress to future link in the curriculum
Systems architecture	<i>Autumn HT1</i>	<ul style="list-style-type: none"> Understand what the CPU of a computer does. Know what major components of the CPU are. Understand what the CPU of a computer does. Know what major components of the CPU are. Know the stages of the fetch, execute cycle. Know what factors affect the speed of a CPU. Know what major components of the CPU are. Know what is meant by the term: ‘embedded system’. Know several examples of embedded systems. 	<i>Year 8: Understanding computers</i>	<i>Year 10: Memory and storage</i> <i>Year 10: Computer networks, protocols and layers</i> <i>Year 10: Hardware and software</i> <i>Year 10: Data representation</i> <i>Year 11: Classification of programming languages</i> A Level: Components of a computer
Memory and storage	<i>Autumn HT1</i>	<ul style="list-style-type: none"> Understand the different types of memory within a computer: RAM, ROM, Cache, Registers. Know the difference between RAM and ROM. Know the purpose of ROM in a computer system. Know the purpose of RAM in a computer system. Understand the need for secondary storage. Know the common types of storage. Know the characteristics of storage devices. Understand the suitability of storage devices for given applications. Understand the advantages and disadvantages of devices based on their characteristics. Know what is meant by cloud storage. Understand the advantages and disadvantages of cloud storage. Know the characteristics of storage devices. 	<i>Year 8: Understanding computers</i>	<i>Year 10: Hardware and software</i> <i>Year 10: Data representation</i> <i>Year 11: Classification of programming languages</i>
Python programming	<i>Autumn HT1</i>	<ul style="list-style-type: none"> Using the Python programming environment and recapping the basics: simple inputs, outputs, and calculations. Using variables and correct data types. Using sequencing in Python. Using selection and iteration in Python: (IF statements, FOR loops and WHILE loops). Using Boolean and arithmetic operators. 	<i>Year 7: Introduction to coding with Kodu</i> <i>Year 7 and 8: Scratch</i> <i>Year 8 and 9: Python programming</i>	<i>Year 10: Python programming</i> <i>Year 10: Basic programming concepts</i>
Computer networks, protocols and layers	<i>Autumn HT1 and HT2</i>	<ul style="list-style-type: none"> Know what is meant by a ‘computer network’. Know the different types of networks: LAN, WAN and PAN. Understand the advantages of networking. Understand the disadvantages of networking. Understand that networks can be wired or wireless. Understand that wired networks typically use copper or fibre optic cables. Understand the advantages of wireless networks. Understand the disadvantages of wireless networks. Know what a star network is. Know what a bus network is. Know the advantages and disadvantages of star and bus networks. Know what is meant by the term “network protocol”. Explain what the Ethernet protocol family is. Describe the use of a range of common network protocols, including TCP, UDP, IP, HTTP, HTTPS, FTP, SMTP and IMAP. Know why protocols are layered. 	<i>Year 7: Networks</i> <i>Year 9: Cybersecurity</i>	<i>Year 10: Cybersecurity</i> A Level: Networks and web technologies

		<ul style="list-style-type: none"> Understand the four layers which make up the TCP/IP protocol model. Understand which common network protocols from last lesson operate at each layer. Understand the need for, and importance of, network security. Explain various methods of network security. 		
Cybersecurity	<i>Autumn HT2</i>	<ul style="list-style-type: none"> Define the term cybersecurity and describe the main purpose of cybersecurity. Understand and be able to explain a range of cybersecurity threats. Define the term social engineering. Describe what social engineering is and how it can be protected against. Explain the following form of social engineering: blagging (pretexting). Explain the following form of social engineering: phishing. Define the term social engineering. Describe what social engineering is and how it can be protected against. Explain the following form of social engineering: shouldering (or shoulder surfing). Explain the following cybersecurity threat: pharming. Define the term malware. Describe what malware is and how it can be protected against. Describe the following form of malware: computer virus. Define the term malware. Describe what malware is and how it can be protected against. Describe the following form of malware: trojan Describe the following form of malware: spyware Explain what penetration testing is and what it is used for. Explain the difference between black-box and white-box penetration testing. Understand the common mistakes people make as the weak points in a system. Understand and be able to explain a range of security measures, including biometrics, password systems, CAPTCHA, Email conformation, Automatic software updates 	<i>Year 9: cybersecurity</i>	<i>Year 10: Ethical, legal and environmental impacts</i>
Python programming	<i>Autumn HT2</i>	<ul style="list-style-type: none"> Using simple functions in Python, making programs more efficient Regular expressions (using regular expressions to validate data) 	<i>Year 7: Introduction to coding with Kodu</i> <i>Year 7 and 8: Scratch</i> <i>Year 8 and 9: Python programming</i> <i>Year 10 HT1: Python programming</i>	<i>Year 10: Python programming</i> <i>Year 10: Basic programming concepts</i>
Hardware and software	<i>Autumn HT2 and Spring HT3</i>	<ul style="list-style-type: none"> Define the term hardware and software. Understand the relationship between hardware and software. Explain what is meant by system software and application software. Understand the functions of operating systems. Understand how the operating system handles the management of the processor(s). Understand how the operating system handles. the management of the memory. Understand how the operating system handles the management of the I/O devices. Understand how the operating system handles the management of the applications. 	<i>Year 8: Understanding computers</i> <i>Year 10: Computer networks, protocols and layers</i>	<i>Year 10: Data representation</i> <i>Year 11: Classification of programming languages</i> <i>A Level: System software</i>

		<ul style="list-style-type: none"> Understand how the operating system handles the management of user security. Understand encryption utilities. Understand defragmentation utilities. Understand data compression utilities. Understand backup utilities. 		
Ethical, legal and environmental impacts	<i>Spring HT3</i>	<ul style="list-style-type: none"> Know a range of things to consider beyond development when implementing new computer systems. Understand at least one ethical issue of digital technology. Understand at least one issue related to privacy and digital technologies. Know the principles of the Acts of Parliament: Data Protection Act 2018; Computer Misuse Act 1990; Copyright Designs and Patents Act 1988; Freedom of Information Act 2000 Understand some of the legal impacts of computer science and its related technologies. Understand the environmental impact of digital technology in terms of: Manufacturing, use and disposal Understand a range of impacts and risks of digital technology Understand how to consider a scenario from the perspective of different groups. Understand at least one scenario of the impact and risks of digital technology. 	<i>Year 7: Impact of technology</i> <i>Year 9: Cybersecurity</i> <i>Year 10: Cybersecurity</i> <i>Year 10: Computer networks, protocols and layers</i>	<i>Examination revision</i> <i>A Level: Ethical, legal and environmental impacts</i>
Python programming	<i>Spring HT3</i>	<ul style="list-style-type: none"> Using lists, edit and appending lists Sorting lists 	<i>Year 7: Introduction to coding with Kodu</i> <i>Year 7 and 8: Scratch</i> <i>Year 8 and 9: Python programming</i> <i>Year 10 HT1 and HT2: Python programming</i>	<i>Year 10: Python programming</i> <i>Year 10: Basic programming concepts</i>
Data representation	<i>Spring HT4</i>	<ul style="list-style-type: none"> Understand why computers systems use binary to store data. Understand that computers use binary to represent many different types of data and instructions. Understand the number bases: Decimal (base 10); binary (base 2); hexadecimal (base 16) Explain why hexadecimal is often used in computer science. Understand how binary can be used to represent whole numbers. Under how hexadecimal can be used to represent whole numbers. Know how to convert positive decimal whole numbers (0–255) into 8-bit binary numbers and vice versa. Know how to convert positive binary whole numbers (0-255) into hexadecimal. Know how to convert positive decimal whole numbers (0-255) into hexadecimal. Know how to add together up to three 8-bit binary numbers. Know how to perform a left and right binary shift. Understand what binary shift achieves. Understand what is meant by the terms bit, nibble, byte, kilobyte, megabyte, gigabyte and terabyte. Know how to represent the capacity of data storage using these units and be able to convert between them. Understand that all data must be represented in binary numbers, including text. Know what is meant by the term “character set”. Understand the relationship between the number of bits in the character set and the number of characters that can be represented. Know two common character sets: ASCII and Unicode. 	<i>Year 8: Understanding computers</i> <i>Year 8: Media (Vector graphics)</i> <i>Year 9: Graphics</i> <i>Year 9: Sound editing in Audacity</i> <i>Year 9: Media (Animations)</i>	<i>Year 10: Advanced programming techniques</i> <i>Year 11: Classification of programming languages</i> <i>Year 11: Relational databases and SQL</i> <i>Year 11: Boolean logic</i> <i>A Level: Exchanging data</i>

		<ul style="list-style-type: none"> Understand how an image is represented as a series of pixels represented in binary. Explain what the terms “size of pixels” and “colour depth” mean. Explain how to calculate the physical size of a bitmap image. Understand the effect of colour depth and pixel dimensions on the size of an image file. Be able to covert binary data into a simple image and vice versa. Understand how sound can be sampled and stored in digital form. Understand the difference between analogue and digital sound. Understand how sampling rate and sample resolution affect the size of a sound file and the quality of its playback. Know why data is often compressed for transfer and storage. Understand the difference between lossy and lossless compression. Explain how data can be compressed using Huffman coding. Be able to interpret and create Huffman trees. Be able to calculate the rate of compression using Huffman trees. Know why data is often compressed for transfer and storage. Understand the difference between lossy and lossless compression. Explain how data can be compressed using run-length encoding (RLE). Be able to represent data in RLE frequency/data pairs. 		
Python programming	<i>Spring HT4</i>	<ul style="list-style-type: none"> Reading and writing files Using two dimensional lists 	<i>Year 7: Introduction to coding with Kodu Year 7 and 8: Scratch Year 8 and 9: Python programming Year 10 HT1, HT2 and HT3: Python programming</i>	<i>Year 11: Algorithms Year 11: Boolean logic A Level: programming</i>
Basic programming concepts	<i>Summer HT5</i>	<ul style="list-style-type: none"> Know the different variable data types. Know the arithmetic operators. Know the Boolean operators. Know the comparison operators. Understand what a variable is and when to use one. Understand what a constant is and when to use one. Understand how to get input from the keyboard. Understand how to output data to the screen. Understand why it is important to use meaningful identifier names. Know the three programming constructs: Sequence, selection, and iteration Understand the use of nested selection and iteration structures. Understand that programs can be structured using procedures and functions. 	<i>Year 7: Introduction to coding with Kodu Year 7 and 8: Scratch Year 8 and 9: Python programming Year 9: Computational thinking Year 10 HT1, HT2 and HT3: Python programming</i>	<i>Year 10: Advanced programming concepts Year 10: Robust and secure programming concepts Year 11: Algorithms Year 11: Boolean logic Year 11: Classification of programming languages Year 11: Programming examination revision A Level programming</i>
Advanced programming concepts	<i>Summer HT6</i>	<ul style="list-style-type: none"> Understand the concept of a data structure. Understand how an array or list can be used to store data. Understand that arrays can be one or two dimensional. Understand how to use records to help solve simple problems. Understand and be able to use a range of string manipulation functions. Understand how to convert strings into other data types and vice versa. Understand how to use random number generation. Understand how to use subroutines. Explain the advantages of using subroutines. 	<i>Year 7: Introduction to coding with Kodu Year 7 and 8: Scratch Year 8 and 9: Python programming Year 9: Computational thinking Year 10 HT1, HT2 and HT3: Python programming Year 10: Basic programming concepts</i>	<i>Year 10: Robust and secure programming concepts Year 11: Algorithms Year 11: Boolean logic Year 11: Classification of programming languages Year 11: Relational databases and SQL Year 11: Programming examination revision A Level programming</i>

		<ul style="list-style-type: none"> • Describe the user of parameters to pass data within programs. • Understand that subroutines can return values to the calling routine. • Understand how to use subroutines. • Understand how to use local variables. • Explain the importance of using local variables. • Describe the structured approach to programming • Explain the advantages of the structured approach to programming. 		
Robust and secure programming	<i>Summer HT6</i>	<ul style="list-style-type: none"> • Understand why data validation is necessary. • Know a range of validation techniques that can be used to write simple data validation routines. • Understand some authentication techniques a programmer may choose to use to protect their program from misuse. • Be able to write simple authentication routines. • Understand that because a program works, it doesn't mean it works for all inputs. • Understand that suitable test data for a program needs to include: Normal (typical) data, boundary (extreme) data, erroneous data • Be able to state four reasons why a program should be tested. • Know what a syntax error is. • Know what a logic error is. • Understand how to identify and fix syntax and logic errors. 	<i>Year 7: Introduction to coding with Kodu</i> <i>Year 7 and 8: Scratch</i> <i>Year 8 and 9: Python programming</i> <i>Year 9: Computational thinking</i> <i>Year 10 HT1, HT2 and HT3: Python programming</i> <i>Year 10: Basic programming concepts</i> <i>Year 10: Advanced programming concepts</i>	<i>Year 11: Algorithms</i> <i>Year 11: Boolean logic</i> <i>Year 11: Classification of programming languages</i> <i>Year 11: Relational databases and SQL</i> <i>Year 11: Programming examination revision</i> <i>A Level: Software development</i>