Computer Science

Key Stage 4

Curriculum Overview

Key Stage 4 Curriculum overview: Computer Science

Y10	Week 1						Week 39
	2.2 Programming techniques	2.1 Algorithms	1.1 Systems Architecture	2.3 Producing robust programs	1.2 memory and storage	2.4 Boolean Logic	1.3 Computer Network connections and protocols
Key content (know that…Know how…)	▪ Basic programming constructs of sequence, selection and iteration are used to control the flow of a program ▪ Common arithmetic operators and Boolean operators of AND, OR, NOT ▪ Use of records to store and search for data ▪ Use of 1D and 2D arrays when solving problems ▪ Use subprograms to produce structured code	▪ Principles of computational thinking ▪ Identify inputs, processes and outputs for a problem ▪ Structure diagrams ▪ Create, interpret, correct, complete and refine algorithms ▪ Identify common errors in algorithms ▪ Complete trace tables ▪ Searching and sorting algorithms	▪ Purpose of the CPU ▪ Common CPU components and their purpose ▪ Von Neumann architecture ▪ Common characteristics of CPUs and how they affect performance ▪ Purpose and characteristics of embedded systems ▪	▪ Defensive design considerations ▪ Input validation ▪ Maintainable programs ▪ Purpose and types of testing programs ▪ Types of errors ▪ Selecting and using suitable test data ▪ Refining algorithms	▪ Purpose of RAM and Rom ▪ Common types of storage ▪ Suitable storage devices and storage media ▪ Units of data storage ▪ Convert positive whole numbers to binary numbers ▪ Use binary codes to represent characters ▪ How an image is represented as a series of pixels ▪ The need for compression	▪ Characteristics and purpose of different levels of programming languages ▪ Purpose of translators (Inc compilers and interpreters) ▪ Common tools and features of an IDE ▪ Simple logic diagrams using AND, OR, NOT ▪ Draw and complete truth tables ▪ Combine Boolean operators	▪ Types of networks ▪ Factors that affect the performance of networks ▪ Different roles of computers in a client server and peer to peer network ▪ Hardware needed to connect stand alone computers into a local area network ▪ Modes of connection ▪ Encryption ▪ Understand the concept of layers

Prior Knowledge	Ks3 : use two or more programming languages KS3: Computational thinking : abstraction KS3: Computational thinking : outputting data KS3: Microbit: user input, selection, iteration	KS3: design, use and evaluate computational abstractions understand several key algorithms that reflect computational thinking KS3: Computational thinking : abstraction, decomposition KS3: Computer systems – sequencing instructions KS3: Searching and sorting algorithms	KS3: Understand the hardware and software components that make up computer systems KS3: Understand how instructions are stored and executed within a computer system KS3: PC basics – input, output, storage KS3 – PC Basics – primary and secondary storage KS3 – PC Basics – CPU	2.1 algorithms – refining algorithms KS3: Python KS3: ICT project spreadsheets	KS3: Understand the hardware and software components that make up computer systems KS3: understand how numbers can be represented in binary 1.1 Common CPU components and their purpose	2.2 programming fundamentals 2.3 producing robust programs 2.1 Algorithms Ks3 : understand simple Boolean logic [for example, AND, OR and NOT]	1.1 common characteristics of CPUs and how they affect performance KS3: Understand the hardware and software components that make up computer systems
GCSE Assessment Objectives	AO1: Demonstrate knowledge and understanding of the key concepts and principles of computer science AO2: Apply knowledge and understanding of key concepts and principles of computer science AO3: Analyse problems in computational terms	AO1: Demonstrate knowledge and understanding of the key concepts and principles of computer science AO2: Apply knowledge and understanding of key concepts and principles of computer science AO3: Analyse problems in computational terms	AO1: Demonstrate knowledge and understanding of the key concepts and principles of computer science AO2: Apply knowledge and understanding of key concepts and principles of computer science	AO1: Demonstrate knowledge and understanding of the key concepts and principles of computer science AO2: Apply knowledge and understanding of key concepts and principles of computer science AO3: Analyse problems in computational term	AO1: Demonstrate knowledge and understanding of the key concepts and principles of computer science AO2: Apply knowledge and understanding of key concepts and principles of computer science	AO1: Demonstrate knowledge and understanding of the key concepts and principles of computer science AO2: Apply knowledge and understanding of key concepts and principles of computer science	AO1: Demonstrate knowledge and understanding of the key concepts and principles of computer science AO2: Apply knowledge and understanding of key concepts and principles of computer science

Assessments	2.2 programming fundamentals assessment *Key knowledge assessed ; programming fundamentals, data types, basic string manipulation, SQL :*	2.1 Algorithm assessment *Key knowledge assessed: computational thinking, designing, creating and refining an algorithm, searching and sorting algorithms*	1.1 Systems Architecture assessment *Key Knowledge assessed : architecture of the CPU, CPU performance, embedded systems,*	2.3 producing robust programs assessment *Key knowledge assessed : defensive design, testing, identifying syntax and logic errors, refining algorithms*	1.2 memory and storage assessment *Key knowledge assessed : primary memory, secondary memory, units of data, data storage – characters, images and sound*	2.4 programming languages and Boolean logic assessment *Key knowledge assessed : logic diagrams, truth tables, combining Boolean operators, applying logical operators in truth tables to solve problems*	1.3 Computer networks assessment *Key knowledge assessed: networks and topologies, hardware needed to connect to networks, wired and wireless networks, protocols and layers*

Y11	Week 1					Week 39

	1.4 Network security and 1.5 systems software	*2.5 Programming Languages and IDE's*	*1.6 Ethical, Legal and cultural concerns*	*Practical programming*	*Paper 1 revision*	*Paper 2*
Key content (know that…Know how…)	▪ Threats posed to devices/systems ▪ Knowledge/principles of each form of attack including how the attack is used and the purpose of the attack ▪ Understand how to limit the threats posed ▪ Identify methods of vulnerabilities and how it limited the attack ▪ Purpose and functionality of operating systems ▪ Purpose and functionality of utility software	▪ characteristics and purpose of different levels of programming languages ▪ Purpose of translators (Inc compilers and interpreters) ▪ Common tools and features of an IDE	▪ Impacts of digital technology on wider society ▪ Legislation relevant to computer science	▪ Use of variables, constants, operators, inputs, outputs and assignments ▪ Three basic programming constructs (sequence/selection/iteration) ▪ Common arithmetic operators ▪ Common Boolean operators ▪ Random number generation	▪ Systems architecture ▪ Memory and storage ▪ Computer networks, connections and protocols ▪ Network security ▪ Systems security ▪ Ethical, legal, cultural and environmental impacts of digital technology	▪ Algorithms ▪ Programming fundamentals ▪ Producing robust programs ▪ Boolean logic ▪ Programming languages and integrated development environments ▪
Prior Knowledge	KS3: Understand a range of ways to use technology safely, respectfully, responsibly and securely KS3: Understand the hardware and software components that make up computer systems KS3 – networks – LAN/WAN/ hardware and topologies	2.2 programming fundamentals 2.3 producing robust programs 2.1 Algorithms Ks3 : understand simple Boolean logic [for example, AND, OR and NOT] KS3 – programming – selection,	KS3: Understand a range of ways to use technology safely, respectfully, responsibly and securely KS3 – e-safety – right to access, data protection, right to privacy	KS3: Use two or more programming languages KS3: Make appropriate use of data structures KS3:Understand simple Boolean (AND, OR, NOT) and some of its use in circuits KS3 - programming., microbit	KS3 – PC basics, Networks, Binary, services provided by IT	KS3 – Microbit, computational thinking, programming, python

		iteration, outputting data	KS3 – services provided by IT			
GCSE Assessment Objectives	AO1: Demonstrate knowledge and understanding of the key concepts and principles of computer science AO2: Apply knowledge and understanding of key concepts and principles of computer science	AO1: Demonstrate knowledge and understanding of the key concepts and principles of computer science AO2: Apply knowledge and understanding of key concepts and principles of computer science	AO1: Demonstrate knowledge and understanding of the key concepts and principles of computer science AO2: Apply knowledge and understanding of key concepts and principles of computer science	AO1: Demonstrate knowledge and understanding of the key concepts and principles of computer science AO2: Apply knowledge and understanding of key concepts and principles of computer science AO3: Analyse problems in computational terms	AO1: Demonstrate knowledge and understanding of the key concepts and principles of computer science AO2: Apply knowledge and understanding of key concepts and principles of computer science AO3: Analyse problems in computational terms	AO1: Demonstrate knowledge and understanding of the key concepts and principles of computer science AO2: Apply knowledge and understanding of key concepts and principles of computer science AO3: Analyse problems in computational terms
Assessments	1.4 Network security assessment 1.5 system software assessment *Key knowledge assessed : threats to computer systems and networks, identifying and preventing*	2.4 programming languages and Boolean logic assessment *Key knowledge assessed – truth tables for each logic gate, recognition of*	1.6 Ethical, Legal and Cultural assessment *Key knowledge assessed : impacts of digital technology on*	Programming fundamentals assessment *Key knowledge assessed – design, write, test and refine programs.*		

	vulnerabilities, forms of attack, common prevention methods, purpose and functionality of operating systems, utility software,	*the gate symbol create or edit logic diagrams, using notations*	*wider society, legislation relevant to computer science*			