	Unit Title		
Half term	SOW	Description	Skills & content covered
10.1 and 1.1 10.2 Syst Arcl and Mer Algo 2.4 Con al lo Data Rep on	1.1 and 1.2 Systems Architecture and Memory. 2.1	Description of CPU, Memory, Algorithms, Computational Logic and Data	The purpose of the CPU - Von Neumann architecture: MAR (Memory Address Register) MDR (Memory Data Register) Program Counter Accumulator • common CPU components and their function: ALU (Arithmetic Logic Unit) CU (Control Unit) Cache • the function of the CPU as fetch and execute instructions stored in memory • how common characteristics of CPUs affect their performance: Oclock speed Ocache size Onumbe of cores • embedded systems: Opurpose of embedded systems Oexamples of embedded systems. Memory • th difference between RAM and ROM • the purpose of ROM in a computer system • the purpose of RAM in a computer system. • Memory • flash memory. Algorithms using: - pseudocode - using flow diagrams • interpret, correct or complete algorithms Computational logic- why data is represented in computer systems in binary form • simple logic diagrams using operations AND, OR and NOT • truth tables • combining Boolean operators using AND, OR and NOT to two levels applying logical operators in appropriate truth tables to solve problems. Data representation - bit nibble, byte, kilobyte, megabyte, gigabyte, terabyte, petabyte • how data needs to be converted into a binary format to be processed by a computer. Numbers • how to convert positive denary whole numbers (0-255) into 8 bit binary numbers and vice versa • how to add two 8 bit binary integers and explain overflow errors which may occur • bir shifts • how to convert positive denary whole numbers (0-255) into 2 digit hexadecimal numbers and vice versa • how to contert from binary to hexadecimal equivalents and vice versa • check digits. Characters • the use of the three bas programming constructs used to control the flow of a program - sequence • sequence • selection Octave to solve problems. Jourgam: - sequence • Selection • Creater and the number of characters • the terd of a program - sequence • caracter and string • casting • the common arithmetic operators. • the common arithmetic operators. • the use of basic string manipulation the use of arrays (or equivalent) wh
	Algorithms, 2.4 Computation al logic, 2.6 Data Representati on	Representation	
		Programming techniques revisited from Year 9 - PG Online Python next steps	
	2.2 Programming Techniques		 • how to use sub programs (functions and procedures) to produce structured code

10.3 and	1.3 storage.	Description of	Storage - the need for secondary storage
10 4	2 6 Data	Storage Data	data capacity and calculation of data capacity requirements common types of storage: -optical - magnetic -solid
10.1	Donnocontoti	Depresentation	state • suitable storage devices and storage media for a given application, and the advantages and disadvantages of these using characteristics: - canacity - speed - nortability - durability - reliability - cost Data representation - Images
	Representati	Representation	 how an image is represented as a series of pixels represented in binary mage is represented as a series of pixels represented in binary
	on, 1.4	continued,	of colour depth and resolution on the size of an image file. Sound • how sound can be sampled and stored in digital
	Wired and	Wired and	form • how sampling intervals and other factors affect the size of a sound file and the quality of its playback: -
	Wireless	wireless	sample size - bit rate - sampling frequency. Compression • need for compression • types of compression: - lossy - lossless. Wired and wireless networks - • types of networks: - LAN (Local Area Network)- WAN (Wide Area Network)
	Networks,	networks,	• factors that affect the performance of networks • the different roles of computers in a client-server and a peer-to-
	1.5 Network	network	peer network • the hardware needed to connect stand-alone computers into a Local Area Network: - wireless access
	topologies,	protocols.	worldwide collection of computer networks: - DNS (Domain Name Server - hosting - the cloud• the concept of
	protocols		virtual networks. Networks and network protocols - star and mesh network topologies • Wifi:- frequency and
	and lavers		channels - encryption• ethernet• the uses of IP addressing, MAC addressing, and protocols including: - TCP/IP
	ana layers	Writing to filos	Transfer Protocol Secure) - FTP (File Transfer Protocol) - POP (Post Office Protocol) - IMAP (Internet Message Access
	22	writing to mes,	Protocol) - SMTP (Simple Mail Transfer Protocol) • the concept of layers • packet switching.
	Drogromming	continued	
	Programming	practise whilst	Programming techniques - the use of basic file handling operations: - open - read - write - close. Continual practise of programming techniques - with algorithms written for each solution
	techniques,	writing	
	2.1	algorithms.	
	Algorithms		
10.5 and 6	1.7 Systems	Systems	Systems software - the purpose and functionality of systems software
	Software,	software and	 operating systems: - user interface - memory management/multitasking - peripheral
	1.6 Systems	security	management and drivers - user management - file management $ullet$ utility system software: -
	Security	,	encryption software - defragmentation - data compression - the role and methods of
	Security		backup: full - incremental. Systems Security - forms of attack
			• threats posed to networks: - malware - phishing - people as the 'weak point' in secure
			systems (social engineering) - brute force attacks - denial of service attacks
			- data interception and theft - the concept of SQL $$ - poor network policy $ \bullet $ identifying and
			preventing vulnerabilities:
			⊖ penetration testing
			◯ network forensics
			Onetwork policies
			🔿 anti-malware software

2.2	Using	Continual practise of programming techniques.	Applying mathematical operations including: 🔾+
Programming	mathematical	0-	
Techniques,	operators.	\bigcirc *	
2.3	Creating robust	⊖ MOD	
Producing	programs as		Producing Robust Programs - defensive design considerations:
Robust	part of	 input sanitisation/validation planning for contingencies 	
Programs,	programming	 anticipating misuse 	
2.4	practise.	authentication maintainability:	
Computation		 comments 	
al Logic		 indentation the purpose of testing 	
		• types of testing	
		⊖ iterative	
		○ final/terminal	
		 how to identify syntax and logic errors 	
		 selecting and using suitable test data. 	

	Unit Title		
	Hyperlink		
Half term	to SOW	Description	Skills & content covered
11.1 and 2	to SOW Programming Project preparation. 2.1 - Algorithms, 3.2 - Analysis, 3.3 - Design, 3.4 - Development 3.5 -Testing and Evaluation	Learning about different binary sort techniques. Analysing, designing, developing and testing a programming task, in preparation for actual project.	Algorithms - standard searching algorithms: - binary search - linear search • standard sorting algorithms: - bubble sort - merge sort - insertion sort. Analysis - how to analyse and identify the requirements for a solution to the problem • how to set clear objectives that show an awareness of the need for real world utility • how to use abstraction and decomposition to design the solution to a problem • how to identify the data requirements for their system • how to identify test procedures to be used during and after development to check their system against the success criteria • how to use validation to ensure a robust solution to a problem. Design - how to design suitable algorithms to represent the solution to a problem • how to design suitable input and output formats and navigation methods for their system • how to identify suitable variables and structures with appropriate validation for their system • how to use appropriate data types in their system • how to use functions/sub programmes to produce structured reusable code • how to select suitable techniques for the development of the solution. Development - how to develop a solution to the identified problem using a suitable programming language(s) • how to demonstrate testing and refinement of the code during development • how to explain the solution using suitable annotation and evidence of development • how to use suitable techniques to solve all aspects of the problem • how to take a systematic approach to problem solving • how to deeloy practical techniques in an efficient and logical manner • how to show an understanding of the relevant information by presenting evidence of the development of their solutions • how to show an understanding of the technical terminology/concepts that arise from their investigation through analysis of the data collected • how to use the erminology/concepts surrounding their topic and contained in the information collected correctly when it comes to producing analysis in the supporting script. Testing and Evalua
		controlled	controlled conditions in class - 20 hours
		conditions. No	
	Programming	teacher	
	Project	feedback	

11.3	2.2	Learning final	Programming techniques - the use of records to store data • the use of SQL to search for data,
	Programming	programming	Translators and Facilites of languages - characteristics and purpose of different levels of programming
		techniques.	assembler, a compiler and an interpreter • common tools and facilities available in an integrated
	teachniques,	Learning about	development environment (IDE): - editors - error diagnostics - run-time environment - translators.
	2.5	translators.	Ethical concerns - how to investigate and discuss Computer Science technologies while considering: -
	Translators	Discussions	stakeholders are affected by technologies • environmental impact of Computer Science • cultural
	and Facilities	around ethics	implications of Computer Science • open source vs proprietary software • legislation relevant to
	of languages,	and	Computer Science: - The Data Protection Act - Computer Misuse Act 1990 - Copyright Designs and Patents Act 1988 - Creative Commons Licensing - Freedom of Information Act 2000.
	1.8 Ethical,	environment.	
	legal,		
	cultural and		
	environment		
11.4	REVISION		
11.5	REVISION		