

Topic Code	Part	Sub-topic	2021	2022	2023	2024	2025	Mac's Prediction
1.1		<b>The characteristics of contemporary processors, input, output, and storage devices</b>						
1.1.1	(a)	Structure and function of the processor: The Arithmetic and Logic Unit (ALU), Control Unit and Registers (Program Counter (PC), Accumulator (ACC), Memory Address Register (MAR), Memory Data Register (MDR), Current Instruction Register (CIR)). Buses: data, address, and control: how this relates to assembly language programs.						Overdue a large question on this topic - this could relate to factors affecting the performance as a 9/12 marker.
	(b)	The Fetch-Decode-Execute Cycle; including its effects on registers.						
	(c)	The factors affecting the performance of the CPU: clock speed, number of cores, cache.						
	(d)	The use of pipelining in a processor to improve efficiency.						

	(e)	Von Neumann, Harvard and contemporary processor architecture.					
1.1.2	(a)	The differences between and uses of CISC and RISC processors.		<b>12 Marker</b>			
	(b)	GPUs and their uses (including those not related to graphics).					
	(c)	Multicore and Parallel systems.					
1.1.3	(a)	How different input, output, and storage devices can be applied to the solution of different problems.	<b>9 marker</b>				
	(b)	The uses of magnetic, flash and optical storage devices.					
	(c)	RAM and ROM.					
	(d)	Virtual storage.				<b>9 marker with virtual machines</b>	
<b>1.2</b>		<b>Software and software development</b>					
1.2.1	(a)	The need for, function, and purpose of operating systems.					

(b)	Memory Management (paging, segmentation, and virtual memory).				<b>9 Marker</b>		
(c)	Interrupts, the role of interrupts and Interrupt Service Routines (ISR), role within the Fetch-Decode-Execute Cycle.						
(d)	Scheduling: round-robin, first come first served, multi-level feedback queues, shortest job first and shortest remaining time.						
(e)	Distributed, embedded, multi-tasking, multi-user and Real-Time operating systems.						
(f)	BIOS.						
(g)	Device drivers.						
(h)	Virtual machines, any instance where software is used to take on the function of a machine, including executing intermediate code or running an operating system within another.					<b>9 marker with virtual storage</b>	

1.2.2	(a)	The nature of applications, justifying suitable applications for a specific purpose.						
	(b)	Utilities.						
	(c)	Open source vs closed source.						
	(d)	Translators: Interpreters, compilers, and assemblers.						
	(e)	Stages of compilation (lexical analysis, syntax analysis, code generation, and optimisation).						
	(f)	Linkers and loaders and use of libraries.						
1.2.3	(a)	Understand the waterfall lifecycle, agile methodologies, extreme programming, the spiral model, and rapid application development.						
	(b)	The relative merits and drawbacks of different methodologies and when they might be used.						
	(c)	Writing and following algorithms.						

**9  
Marker**

1.2.4	(a)	Need for and characteristics of a variety of programming paradigms.					9 marker on high/low level languages and their purposes
	(b)	Procedural languages.					
	(c)	Assembly language (including following and writing simple programs with the Little Man Computer instruction set).					
	(d)	Modes of addressing memory (immediate, direct, indirect, and indexed).	12 Marker				
	(e)	Object-oriented languages with an understanding of classes, objects, methods, attributes, inheritance, encapsulation, and polymorphism.		7 Marker			
1.3		<b>Exchanging data</b>					
1.3.1	(a)	Lossy vs Lossless compression.					
	(b)	Run length encoding and dictionary coding for lossless compression.					

	(c)	Symmetric and asymmetric encryption.			9 Marker			
	(d)	Different uses of hashing.			9 Marker			
1.3.2	(a)	Relational database, flat file, primary key, foreign key, secondary key, entity relationship modelling, normalisation, and indexing.				9 Marker		
	(b)	Methods of capturing, selecting, managing, and exchanging data.						
	(c)	Normalisation to 3NF.						
	(d)	SQL – Interpret and modify.						
	(e)	Referential integrity.						
	(f)	Transaction processing, ACID (Atomicity, Consistency, Isolation, Durability), record locking, and redundancy.						
1.3.3	(a)	Characteristics of networks and the importance of protocols and standards.						

	(b)	The internet structure: The TCP/IP Stack, DNS, protocol layering, LANs and WANs, packet and circuit switching.					
	(c)	Network security and threats, use of firewalls, proxies, and encryption.					
	(d)	Network hardware.					
	(e)	Client-server and peer-to-peer.					
1.3.4	(a)	HTML, CSS, and JavaScript.					
	(b)	Search engine indexing.					
	(c)	PageRank algorithm.					
	(d)	Server and client-side processing.					
1.4		<b>Data types, data structures, and algorithms</b>					
1.4.1	(a)	Primitive data types, integer, real/floating point, character, string, and Boolean.					
	(b)	Represent positive integers in binary.					

	(c)	Use of sign and magnitude and two's complement to represent negative numbers in binary.						
	(d)	Addition and subtraction of binary integers.						
	(e)	Represent positive integers in hexadecimal.						
	(f)	Convert positive integers between binary, hexadecimal, and denary.						
	(g)	Representation and normalisation of floating-point numbers in binary.						
	(h)	Floating-point arithmetic, positive and negative numbers, addition, and subtraction.						
	(i)	Bitwise manipulation and masks: shifts, combining with AND, OR, and XOR.						
	(j)	How character sets (ASCII and UNICODE) are used to represent text.						
1.4.2	(a)	Arrays (of up to 3 dimensions), records, lists, tuples.						

	(b)	The following structures to store data: linked-list, graph (directed and undirected), stack, queue, trees						
	(c)	How to create, traverse, add data to, and remove data from the data structures mentioned above (using arrays and procedural programming or an object-oriented approach).						
1.4.3	(a)	Define problems using Boolean logic.						
	(b)	Manipulate Boolean expressions, including the use of Karnaugh maps to simplify Boolean expressions.						
	(c)	Use the following rules to derive or simplify statements in Boolean algebra: De Morgan's Laws, distribution, association, commutation, double negation.						
	(d)	Using logic gate diagrams and truth tables.						
	(e)	The logic associated with D type flip flops, half and full adders.						
1.5		<b>Legal, moral, cultural, and ethical issues</b>						

1.5.1	(a)	The Data Protection Act 1998.					
	(b)	The Computer Misuse Act 1990.	<b>9 Marker</b>				
	(c)	The Copyright Design and Patents Act 1988.					
	(d)	The Regulation of Investigatory Powers Act 2000.			<b>12 Marker</b>		
1.5.2		The individual moral, social, ethical, and cultural opportunities and risks of digital technology: computers in the workforce, automated decision making, artificial intelligence, environmental effects, censorship and the internet, monitor behaviour, analyse personal information, piracy and offensive communications, layout, colour paradigms, and character sets.		<b>9 Marker</b>		<b>9 Marker Layout and Colour Paradigm s</b>	<b>9 marker on AI and legal/mor al implicatio ns of generatin g images</b>