AQA GCSE Computer Science PLC				Name	1 - No Idea 2 - Shaky 3 - OK 4 - Really Secure
Section	**Topic**	**AQA**	**Learning Objective**		
1 - Fundamentals of Algorithms	Representing algorithms	1	Understand and explain the term algorithm.		
		2	Understand and explain the term decomposition.		
		3	Understand and explain the term abstraction.		
		4	Use a systematic approach to problem solving and algorithm creation representing those algorithms using pseudo-code and flowcharts.		
		5	Explain simple algorithms in terms of their inputs, processing and outputs.		
		6	Determine the purpose of simple algorithms		
	Efficiency of algorithms	7	Understand that more than one algorithm can be used to solve the same problem.		
		8	Compare the efficiency of algorithms explaining how some algorithms are more efficient than others in solving the same problem.		
	Searching algorithms	9	Understand and explain how the linear search algorithm works.		
		10	Understand and explain how the binary search algorithm works.		
		11	Compare and contrast linear and binary search algorithms.		
	Sorting algorithms	12	Understand and explain how the merge sort algorithm works.		
		13	Understand and explain how the bubble sort algorithm works.		
		14	Compare and contrast merge sort and bubble sort algorithms.		
2 - Programming	Data types	15	Understand the concept of a data type		
		16	Understand and use the following appropriately: • integer • real • Boolean • character • string.		
	Programming concepts	17	Use, understand and know how the following statement types can be combined in programs:		
			Variable Declaration		
			Constant declaration		
			Assignment		
			Iteration		
			Selection		
			Subroutine (procedure / function)		
		18	Use definite and indefinite iteration, including indefinite iteration with the condition(s) at the start or the end of the iterative structure.		
		19	Use nested selection and nested iteration structures.		
		20	Use meaningful identifier names and know why it is important to use them.		
	Relational operations in a programming language	21	Be familiar with and be able to use: • equal to • not equal to • less than • greater than • less than or equal to • greater than or equal to.		
	Boolean operations in a programming language	22	Be familiar with and be able to use: • NOT • AND • OR.		
	Data structures	23	Understand the concept of data structures.		
		24	Use arrays (or equivalent) in the design of solutions to simple problems.		
		25	Use records (or equivalent) in the design of solutions to simple problems.		
	Input / output and file handling	26	Be able to obtain user input from the keyboard		
		27	Be able to output data and information from a program to the computer display.		
		28	Be able to read/write from/to a text file		
	String handling operations in a programming language	29	Understand and be able to use: • length • position • substring • concatenation • convert character to character code • convert character code to character • string conversion operations.		

2	Random number generation in a programming language	30	Be able to use random number generation.	
	Subroutines (procedures and functions)	31	Understand the concept of subroutines	
		32	Explain the advantages of using subroutines in programs.	
		33	Describe the use of parameters to pass data within programs.	
		34	Use subroutines that return values to the calling routine.	
		35	Know that subroutines may declare their own variables, called local variables, and that local variables usually: • only exist while the subroutine is executing • are only accessible within the subroutine.	
		36	Use local variables and explain why it is good practice to do so	
	Structured programming	37	Describe the structured approach to programming.	
		38	Explain the advantages of the structured approach.	
	Robust and secure programming	39	Be able to write simple data validation routines	
		40	Be able to write simple authentication routines.	
		41	Be able to select suitable test data that covers normal (typical), boundary (extreme) and erroneous data. Be able to justify the choice of test data.	
	Classification of programming languages	42	Know that there are different levels of programming language: • low-level language • high-level language. Explain the main differences between low-level and high-level languages.	
		43	Know that machine code and assembly language are considered to be low-level languages and explain the differences between them.	
		44	Understand the advantages and disadvantages of low-level language programming compared with high-level language programming.	
Fundamentals of Data Representation	Number bases	45	Understand the following number bases: • decimal (base 10) • binary (base 2) • hexadecimal (base 16).	
		46	Understand that computers use binary to represent all data and instructions.	
		47	Explain why hexadecimal is often used in computer science.	
	Converting between number bases	48	Understand how binary can be used to represent whole numbers.	
		49	Understand how hexadecimal can be used to represent whole numbers.	
		50	Be able to convert in both directions between: • binary and decimal • binary and hexadecimal • decimal and hexadecimal.	
	Units of information	51	Know that: • a bit is the fundamental unit of information • a byte is a group of 8 bits.	
		52	Know that quantities of bytes can be described using prefixes. Know the names, symbols and corresponding values for the decimal prefixes: • kilo, 1 kB is 1,000 bytes • mega, 1 MB is 1,000 kilobytes • giga, 1 GB is 1,000 Megabytes • tera, 1 TB is 1,000 Gigabytes.	
	Binary arithmetic	53	Be able to add together up to three binary numbers.	
		54	Be able to apply a binary shift to a binary number.	
		55	Describe situations where binary shifts can be used.	
	Character encoding	56	Understand what a character set is and be able to describe the following character encoding methods: • 7-bit ASCII • Unicode.	
		57	Understand that character codes are commonly grouped and run in sequence within encoding tables.	
		58	Describe the purpose of Unicode and the advantages of Unicode over ASCII. Know that Unicode uses the same codes as ASCII up to 127.	
		59	Understand what a pixel is and be able to describe how pixels relate to an image and the way images are displayed	
		60	Describe the following for bitmaps: • size in pixels • colour depth.	

3 - Fu	Represenging images	61	Describe how a bitmap represents an image using pixels and colour depth.	
		62	Describe using examples how the number of pixels and colour depth can affect the file size of a bitmap image.	
		63	Calculate bitmap image file sizes based on the number of pixels and colour depth.	
		64	Convert binary data into a black and white image.	
		65	Convert a black and white image into binary data.	
	Representing sound	66	Understand that sound is analogue and that it must be converted to a digital form for storage and processing in a computer.	
		67	Understand that sound waves are sampled to create the digital version of sound.	
		68	Describe the digital representation of sound in terms of: • sampling rate • sample resolution.	
		69	Calculate sound file sizes based on the sampling rate and the sample resolution.	
	Data compression	70	Explain how data can be compressed using Huffman coding. Be able to interpret Huffman trees.	
		71	Be able to calculate the number of bits required to store a piece of data compressed using Huffman coding. Be able to calculate the number of bits required to store a piece of uncompressed data in ASCII.	
		72	Explain how data can be compressed using run length encoding (RLE).	
		73	Represent data in RLE frequency/data pairs	
4 - Computer Systems	Hardware and software	74	Define the terms hardware and software and understand the relationship between them.	
	Boolean logic	75	Construct truth tables for the following logic gates: • NOT • AND • OR.	
		76	Construct truth tables for simple logic circuits. Interpret the results of simple truth tables.	
		77	Create, modify and interpret simple logic circuit diagrams.	
	Software classification	78	Explain what is meant by: • system software • application software. Give examples of both types of software.	
		79	Understand the need for, and functions of, operating systems (OS) and utility programs. Understand that the OS handles management of the: • processor(s) • memory • I/O devices • applications • security.	
	Systems architecture	80	Explain the Von Neumann architecture	
		81	Explain the role and operation of main memory and the following major components of a central processing unit (CPU): • arithmetic logic unit • control unit • clock • bus.	
		82	Explain the effect of the following on the performance of the CPU: • clock speed • number of processor cores • cache size • cache type.	
		83	Understand and explain the Fetch-Execute cycle.	
		84	Understand the differences between main memory and secondary storage. Understand the differences between RAM and ROM.	
		85	Understand why secondary storage is required.	
		86	Be aware of different types of secondary storage (solid state, optical and magnetic). Explain the operation of solid state, optical and magnetic storage. Discuss the advantages and disadvantages of solid state, optical and magnetic storage.	
		87	Explain the term 'cloud storage'.	
		88	Explain the advantages and disadvantages of cloud storage when compared to local storage.	
		89	Understand the term 'embedded system' and explain how an embedded system differs from a non-embedded system.	
		90	Define what a computer network is. Discuss the benefits and risks of computer networks.	
		91	Describe the main types of computer network including: • Personal Area Network (PAN) • Local Area Network (LAN) • Wide Area Network (WAN)	

5 - Fundamentals of Computer Networks	Fundamentals of computer networks	92	Understand that networks can be wired or wireless. Discuss the benefits and risks of wireless networks as opposed to wired networks.	
		93	Explain the following common network topologies: • star • bus.	
		94	Define the term ‘network protocol’.	
		95	Explain the purpose and use of common network protocols including: • Ethernet • Wi-Fi • TCP (Transmission Control Protocol) • UDP (User Datagram Protocol) • IP (Internet Protocol) • HTTP (Hypertext Transfer Protocol) • HTTPS (Hypertext Transfer Protocol Secure) • FTP (File Transfer Protocol) • email protocols: • SMTP (Simple Mail Transfer Protocol) • IMAP (Internet Message Access Protocol).	
		96	Understand the need for, and importance of, network security.	
		97	Explain the following methods of network security: • authentication • encryption • firewall • MAC address filtering.	
		98	Describe the 4 layer TCP/IP model: • application layer • transport layer • network layer • data link layer	
		99	Understand that the HTTP, HTTPS, SMTP, IMAP and FTP protocols operate at the application layer.	
		100	Understand that the TCP and UDP protocols operate at the transport layer	
		101	Understand that the IP protocol operates at the network layer.	
6 - Fundamentals of Cyber Security	Fundamentals of cyber security	102	Be able to define the term cyber security and be able to describe the main purposes of cyber security	
	Cyber security threats	103	Understand and be able to explain the following cyber security threats: • social engineering techniques • malicious code • weak and default passwords • misconfigured access rights • removable media • unpatched and/or outdated software.	
		104	Explain what penetration testing is and what it is used for.	
	Social engineering	105	Define the term social engineering.	
		106	Describe what social engineering is and how it can be protected against.	
		107	Explain the following forms of social engineering: • blagging (pretexting) • phishing • pharming • shouldering (or shoulder surfing).	
	Malicious code	108	Define the term 'malware'.	
		109	Describe what malware is and how it can be protected against.	
		110	Describe the following forms of malware: • computer virus • trojan • spyware • adware.	
	Methods to detect and prevent cyber security threats	111	Understand and be able to explain the following security measures: • biometric measures (particularly for mobile devices) • password systems • CAPTCHA (or similar) • using email confirmations to confirm a user’s identity • automatic software updates.	
7 - Ethical, Legal and Environmental issues	Ethical, legal and environmental impacts of digital technology on wider society, including issues of privacy	112	Explain the current ethical, legal and environmental impacts and risks of digital technology on society. Where data privacy issues arise these should be considered.	
amming report	**Design**	113	Be aware that before constructing a solution, the solution should be designed, for example planning data structures for the data model, designing algorithms, designing an appropriate modular structure for the solution and designing the user interface.	
	Implementation	114	Be aware that the models and algorithms need to be implemented in the form of data structures and code (instructions) that a computer can understand.	

NEA: Practical progra	**Testing**	115	Be aware that the implementation must be tested for the presence of errors, using selected test data covering normal (typical), boundary (extreme) and erroneous data.	
	Evaluation / refining	116	Be aware that code created during implementation will often require refining as a result of testing. Be aware of the importance of assessing how well the solution meets the requirements of the problem and how the solution could be improved if the problem were to be revisited.	