

3.1 Fundamentals of algorithms

3.1.1 Representing algorithms	
Content	Additional information
Understand and explain the term algorithm.	An algorithm is a sequence of steps that can be followed to complete a task. Be aware that a computer program is an implementation of an algorithm and that an algorithm is not a computer program.
Understand and explain the term decomposition.	Decomposition means breaking a problem into a number of sub-problems, so that each sub-problem accomplishes an identifiable task, which might itself be further subdivided.
Understand and explain the term abstraction.	Abstraction is the process of removing unnecessary detail from a problem.
Use a systematic approach to problem solving and algorithm creation representing those algorithms using pseudo-code, program code and flowcharts.	Any exam question where students are given pseudo-code will use the AQA standard version. Exam questions will indicate the form of response expected. For example, pseudo-code, program code or a flowchart.
Explain simple algorithms in terms of their inputs, processing and outputs.	Students must be able to identify where inputs, processing and outputs are taking place within an algorithm.
Determine the purpose of simple algorithms.	Students should be able to use trace tables and visual inspection to determine how simple algorithms work and what their purpose is.

3.1.2 Efficiency of algorithms	
Content	Additional information
Understand that more than one algorithm can be used to solve the same problem.	
Compare the efficiency of algorithms explaining how some algorithms are more efficient than others in solving the same problem.	Formal comparisons of algorithmic efficiency are not required. Exam questions in this area will only refer to time efficiency.

AQA GCSE Computer Science 8525 from 2020

3.1.3 Searching algorithms

Content	Additional information
Understand and explain how the linear search algorithm works.	Students should know the mechanics of the algorithm.
Understand and explain how the binary search algorithm works.	Students should know the mechanics of the algorithm.
Compare and contrast linear and binary search algorithms.	Students should know the advantages and disadvantages of both algorithms.

3.1.4 Sorting algorithms

Content	Additional information
Understand and explain how the merge sort algorithm works.	Students should know the mechanics of the algorithm.
Understand and explain how the bubble sort algorithm works.	Students should know the mechanics of the algorithm.
Compare and contrast merge sort and bubble sort algorithms.	Students should know the advantages and disadvantages of both algorithms.

3.2 Programming

Students need a theoretical understanding of all the topics in this section for the paper 1 exam even if the programming language(s) taught does not support all of the topics. Exams will always present algorithms using the current version of the AQA pseudo-code. The document can be found on the AQA website.

Students need a practical understanding of all the topics and skills in this section for the paper 1 exam. When they are writing computer programs in an exam, they will need to ensure they use meaningful identifier names, use suitable data types and that all logic flow is clearly identifiable to examiners.

Exam questions will explicitly state in what form the response needs to be provided. This will be, for example, pseudo-code, program code or a flowchart, and students must respond as instructed. Where pseudo-code is an accepted method of response, students may present their answers to questions in any suitable format and do not need to use the AQA pseudo-code.

3.2.1 Data types	
Content	Additional information
Understand the concept of a data type.	
Understand and use the following appropriately: <ul style="list-style-type: none"> • integer • real • Boolean • character • string. 	Depending on the actual programming language(s) being used, these variable types may have other names. For example, real numbers may be described as float. In exams we will use the general names given opposite.

3.2.2 Programming concepts	
Content	Additional information
Use, understand and know how the following statement types can be combined in programs: <ul style="list-style-type: none"> • variable declaration • constant declaration • assignment • iteration • selection • subroutine (procedure/function). 	The three combining principles (sequence, iteration/repetition and selection/choice) are basic to all high-level imperative programming languages. Students should be able to write programs using these statement types. They should be able to interpret and write algorithms that include these statement types. Students should know why named constants and variables are used.
Use definite (count controlled) and indefinite (condition controlled) iteration, including indefinite iteration with the condition(s) at the start or the end of the iterative structure.	A theoretical understanding of condition(s) at either end of an iterative structure is required, regardless of whether they are supported by the language(s) being used. An example of definite (count controlled) iteration would be: <pre>FOR i ← 1 TO 5 ... Instructions here ... ENDFOR</pre>

AQA GCSE Computer Science 8525 from 2020

	<p>An example of indefinite (condition controlled) iteration with the condition at the start would be:</p> <pre>WHILE NotSolved ... Instructions here ... ENDWHILE</pre> <p>Examples of indefinite (condition controlled) iteration with the condition at the end would be:</p> <pre>REPEAT ... Instructions here ... UNTIL Solved DO ... Instructions here ... WHILE NotSolved</pre>
<p>Use nested selection and nested iteration structures.</p>	<p>An example of nested iteration would be:</p> <pre>WHILE NotSolved ... Instructions here ... FOR i ← 1 TO 5 ... Instructions here ... ENDFOR ... Instructions here ... ENDWHILE</pre> <p>An example of nested selection would be:</p> <pre>IF GameWon THEN ... Instructions here ... IF Score > HighScore THEN ... Instructions here ... ENDIF ... Instructions here ... ENDIF</pre>
<p>Use meaningful identifier names and know why it is important to use them.</p>	<p>Identifier names include names for variables, constants and subroutine names.</p>

3.2.3 Arithmetic operations in a programming language

Content	Additional information
<p>Be familiar with and be able to use:</p> <ul style="list-style-type: none"> • addition • subtraction • multiplication • real division • integer division, including remainders. 	<p>Integer division, including remainders, is usually a two-stage process and uses modular arithmetic: e.g. the calculation $11/2$ would generate the following values: Integer division: the integer quotient of 11 divided by 2 ($11 \text{ DIV } 2$) = 5 Remainder: the remainder when 11 is divided by 2 ($11 \text{ MOD } 2$) = 1</p>

AQA GCSE Computer Science 8525 from 2020

3.2.4 Relational operations in a programming language

Content	Additional information
Be familiar with and be able to use: <ul style="list-style-type: none">• equal to• not equal to• less than• greater than• less than or equal to• greater than or equal to.	Students should be able to use these operators within their own programs and be able to interpret them when used within algorithms. Note that different languages may use different symbols to represent these operators.

3.2.5 Boolean operations in a programming language

Content	Additional information
Be familiar with and be able to use: <ul style="list-style-type: none">• NOT• AND• OR	Students should be able to use these operators, and combinations of these operators, within conditions for iterative and selection structures.

3.2.6 Data structures

Content	Additional information
Understand the concept of data structures.	It may be helpful to set the concept of a data structure in various contexts that students may already be familiar with. It may also be helpful to suggest/demonstrate how data structures could be used in a practical setting.
Use arrays (or equivalent) in the design of solutions to simple problems.	Only one and two-dimensional arrays are required.
Use records (or equivalent) in the design of solutions to simple problems.	An example of a record definition would be: <pre>RECORD Car make : String model : String reg : String price : Real noOfDoors : Integer ENDRECORD</pre>

3.2.7 Input/output

Content	Additional information
Be able to obtain user input from the keyboard.	
Be able to output data and information from a program to the computer display.	

3.2.8 String handling operations in a programming language

Content	Additional information
Understand and be able to use: <ul style="list-style-type: none"> • Length • Position • Substring • Concatenation • convert character to character code • convert character code to character • string conversion operations. 	Expected string conversion operations <ul style="list-style-type: none"> • string to integer • string to real • integer to string • real to string.

3.2.9 Random number generation in a programming language

Content	Additional information
Be able to use random number generation.	Students will be expected to use random number generation within their computer programs. An understanding of how pseudo-random numbers are generated is not required

3.2.10 Structured programming and subroutines (procedures and functions)

Content	Additional information
Understand the concept of subroutines.	Students should know that a subroutine is a named 'out of line' block of code that may be executed (called) by simply writing its name in a program statement.
Explain the advantages of using subroutines in programs.	
Describe the use of parameters to pass data within programs.	Students should be able to use subroutines that require more than one parameter. Students should be able to describe how data is passed to a subroutine using parameters.
Use subroutines that return values to the calling routine.	Students should be able to describe how data is passed out of a subroutine using return values.
Know that subroutines may declare their own variables, called local variables, and that local variables usually: <ul style="list-style-type: none"> • only exist while the subroutine is executing • are only accessible within the subroutine. 	
Use local variables and explain why it is good practice to do so.	
Describe the structured approach to programming.	Students should be able to describe the structured approach including modularised programming, clear well-documented interfaces (local variables, parameters) and return values. Teachers should be aware that the terms arguments and parameters are sometimes

AQA GCSE Computer Science 8525 from 2020

	used but in examinable material we will use the term parameter to refer to both of these.
Explain the advantages of the structured approach.	
3.2.11 Robust and secure programming	
Content	Additional information
Be able to write simple data validation routines.	<p>Students should be able to use data validation techniques to write simple routines that check the validity of data being entered by a user.</p> <p>The following validation checks are examples of simple data validation routines:</p> <ul style="list-style-type: none"> • checking if an entered string has a minimum length • checking if a string is empty • checking if data entered lies within a given range (e.g. between 1 and 10).
Be able to write simple authentication routines.	Students should be able to write a simple authentication routine that uses a username and password. Students will only be required to use plain text usernames and passwords (ie students will not need to encrypt the passwords).
Understand what is meant by testing in the context of algorithms and programs. Be able to correct errors within algorithms and programs.	
Understand what test data is and describe the following types of test data: <ul style="list-style-type: none"> • normal (typical) • boundary (extreme) • erroneous data. 	Boundary data would be for example: If the allowed range is 1 to 10, then boundary data is 0, 1, 10, 11, i.e. either side of the allowed boundary.
Be able to select and justify the choice of suitable test data for a given problem.	
Understand that there are different types of error: <ul style="list-style-type: none"> • syntax error • logic error. 	
Be able to identify and categorise errors within algorithms and programs.	

3.3 Fundamentals of data representation

3.3.1 Number bases	
Content	Additional information
Understand the following number bases: <ul style="list-style-type: none"> • decimal (base 10) • binary (base 2) • hexadecimal (base 16). 	
Understand that computers use binary to represent all data and instructions.	Students should be familiar with the idea that a bit pattern could represent different types of data including text, image, sound and integer.
Explain why hexadecimal is often used in computer science.	

3.3.2 Converting between number bases	
Content	Additional information
Understand how binary can be used to represent whole numbers.	Students must be able to represent decimal values between 0 and 255 in binary.
Understand how hexadecimal can be used to represent whole numbers.	Students must be able to represent decimal values between 0 and 255 in hexadecimal.
Be able to convert in both directions between: <ul style="list-style-type: none"> • binary and decimal • binary and hexadecimal • decimal and hexadecimal. 	The following equivalent maximum values will be used: <ul style="list-style-type: none"> • decimal: 255 • binary: 1111 1111 • hexadecimal: FF

3.3.3 Units of information	
Content	Additional information
Know that: <ul style="list-style-type: none"> • a bit is the fundamental unit of information • a byte is a group of 8 bits. 	A bit is either a 0 or a 1. <ul style="list-style-type: none"> • b represents bit • B represents byte
Know that quantities of bytes can be described using prefixes. Know the names, symbols and corresponding values for the decimal prefixes: <ul style="list-style-type: none"> • 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. Be able to compare quantities of bytes using the prefixes above.	Students might benefit from knowing that historically the terms kilobyte, megabyte, etc have often been used to represent powers of 2. The International System of Units (SI units) kilo, mega and so forth refer to values based on powers of 10. When referring to powers of 2 the terms kibi, mebi and so forth would normally be used but students do not need to know these.

AQA GCSE Computer Science 8525 from 2020

3.3.4 Binary arithmetic	
Content	Additional information
Be able to add together up to three binary numbers.	Students will need to be able to add together up to three binary numbers using a maximum of 8 bits per number. Students will only be expected to add together a maximum of three 1s in a single column. Answers will be a maximum of 8 bits in length and will not involve carrying beyond the 8th bit.
Be able to apply a binary shift to a binary number.	Students will be expected to use a maximum of 8 bits. Students will be expected to understand and use only a logical binary shift. Students will not need to understand or use fractional representations.
Describe situations where binary shifts can be used.	Binary shifts can be used to perform simple multiplication/division by powers of 2.

3.3.5 Character encoding	
Content	Additional information
Understand what a character set is and be able to describe the following character encoding methods: <ul style="list-style-type: none"> • 7-bit ASCII • Unicode. 	Students should be able to use a given character encoding table to: <ul style="list-style-type: none"> • convert characters to character codes • convert character codes to characters.
Understand that character codes are commonly grouped and run in sequence within encoding tables.	Students should know that character codes are grouped and that they run in sequence. For example, in ASCII 'A' is coded as 65, 'B' as 66, and so on, meaning that the codes for the other capital letters can be calculated once the code for 'A' is known. This pattern also applies to other groupings such as lower-case letters and digits.
Describe the purpose of Unicode and the advantages of Unicode over ASCII. Know that Unicode uses the same codes as ASCII up to 127.	Students should be able to explain the need for data representation of different alphabets and of special symbols allowing a far greater range of characters. It is not necessary to be familiar with UTF-8, UTF-16 or other different versions of Unicode

3.3.6 Representing images	
Content	Additional information
Understand what a pixel is and be able to describe how pixels relate to an image and the way images are displayed.	Students should know that the term pixel is short for Picture Element. A pixel is a single point in an image.

AQA GCSE Computer Science 8525 from 2020

Describe the following for bitmaps: <ul style="list-style-type: none"> • image size • colour depth. <p>Know that the size of a bitmap image is measured in pixels (width x height).</p>	The size of an image is expressed directly as width of image in pixels by height of image in pixels using the notation width x height. Colour depth is the number of bits used to represent each pixel.
Describe how a bitmap represents an image using pixels and colour depth.	Students should be able to explain how bitmaps are made from pixels.
Describe using examples how the number of pixels and colour depth can affect the file size of a bitmap image.	Students should be able to describe how higher numbers of pixels and higher colour depths can affect file size, and should also be able to use examples.
Calculate bitmap image file sizes based on the number of pixels and colour depth.	Students only need to use colour depth and number of pixels within their calculations. $\text{Size (bits)} = W \times H \times D$ $\text{Size (bytes)} = (W \times H \times D) / 8$ $W = \text{image width}$ $H = \text{image height}$ $D = \text{colour depth in bits.}$
Convert binary data into a bitmap image.	Given a binary pattern that represents a simple bitmap, students should be able to draw the resulting image as a series of pixels.
Convert a bitmap image into binary data.	Given a simple bitmap, students should be able to write down a bit pattern that represents the image.

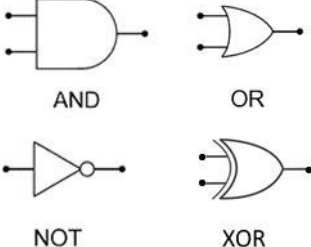
3.3.7 Representing sound

Content	Additional information
Understand that sound is analogue and that it must be converted to a digital form for storage and processing in a computer.	
Understand that analogue signals are sampled to create the digital version of sound.	Students should understand that a sample is a measure of amplitude at a point in time.
Describe the digital representation of sound in terms of: <ul style="list-style-type: none"> • sampling rate • sample resolution. 	Sampling rate is the number of samples taken in a second and is usually measured in hertz (1 hertz = 1 sample per second). Sample resolution is the number of bits per sample.
Calculate sound file sizes based on the sampling rate and the sample resolution.	File size (bits) = rate x res x secs rate = sampling rate res = sample resolution secs = number of seconds

AQA GCSE Computer Science 8525 from 2020

3.3.8 Data compression	
Content	Additional information
<p>Explain what data compression is. Understand why data may be compressed and that there are different ways to compress data.</p>	<p>Students should understand that it is common for data to be compressed and should be able to explain why it may be necessary or desirable to compress data.</p>
<p>Explain how data can be compressed using Huffman coding. Be able to interpret Huffman trees.</p>	<p>Students should be familiar with the process of using a tree to represent the Huffman code.</p>
<p>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.</p>	<p>Students should be familiar with carrying out calculations to determine the number of bits saved by compressing a piece of data using Huffman coding.</p>
<p>Explain how data can be compressed using run length encoding (RLE).</p>	<p>Students should be familiar with the process of using frequency/data pairs to reduce the amount of data stored.</p>
<p>Represent data in RLE frequency/data pairs.</p>	<p>Students could be given a bitmap representation and they would be expected to show the frequency and value pairs for each row, eg 0000011100000011 would become 5 0 3 1 6 0 2 1.</p>

3.4 Computer systems

3.4.1 Hardware and software	
Content	Additional information
Define the terms hardware and software and understand the relationship between them.	
3.4.2 Boolean logic	
Content	Additional information
Construct truth tables for the following logic gates: <ul style="list-style-type: none"> • NOT • AND • OR • XOR. 	Students do not need to know about or use NAND and NOR logic gates.
Construct truth tables for simple logic circuits using combinations of NOT, AND, OR and XOR gates. Interpret the results of simple truth tables.	Students should be able to construct truth tables which contain up to three inputs.
Create, modify and interpret simple logic circuit diagrams. Students will only need to use NOT, AND, OR and XOR gates within logic circuits. Students will be expected to understand and use the following logic circuit symbols: <div style="text-align: center;">  <p>AND OR</p> <p>NOT XOR</p> </div>	Students should be able to construct simple logic circuit diagrams which contain up to three inputs.
Create and interpret simple Boolean expressions made up of NOT, AND, OR and XOR operations.	Students will be expected to understand and use the following Boolean expression operators: <ul style="list-style-type: none"> • is used to represent the AND gate + is used to represent the OR gate ⊕ is used to represent the XOR gate An overbar is used to represent the NOT gate For example: (A AND B) OR NOT C Would use the notation: $A \cdot B + \bar{C}$

AQA GCSE Computer Science 8525 from 2020

<p>Create the Boolean expression for a simple logic circuit. Create a logic circuit from a simple Boolean expression.</p>	
---	--

3.4.3 Software classification	
Content	Additional information
<p>Explain what is meant by:</p> <ul style="list-style-type: none"> • system software • application software. <p>Give examples of both types of software.</p>	
<p>Understand the need for, and functions of, operating systems (OS) and utility programs. Understand that the OS handles management of the:</p> <ul style="list-style-type: none"> • processor(s) • memory • input/output (I/O) devices • applications • security. 	

3.4.4 Classification of programming languages and translators	
Content	Additional information
<p>Know that there are different levels of programming language:</p> <ul style="list-style-type: none"> • low-level language • high-level language. <p>Explain the main differences between low-level and high-level languages.</p>	<p>Students should understand that most computer programs are written in high-level languages and be able to explain why this is the case.</p>
<p>Know that machine code and assembly language are considered to be low-level languages and explain the differences between them.</p>	<p>Students should be able to</p> <ul style="list-style-type: none"> • understand that processors execute machine code and that each type of processor has its own specific machine code instruction set • understand that assembly language is often used to develop software for embedded systems and for controlling specific hardware components • understand that assembly language has a 1:1 correspondence with machine code.
<p>Understand that all programming code written in high-level or assembly languages must be translated. Understand that machine code is expressed in binary and is specific to a processor or family of processors.</p>	

AQA GCSE Computer Science 8525 from 2020

<p>Understand the advantages and disadvantages of low-level language programming compared with high-level language programming.</p>	
<p>Understand that there are three common types of program translator:</p> <ul style="list-style-type: none"> • interpreter • compiler • assembler. <p>Explain the main differences between these three types of translator.</p> <p>Understand when it would be appropriate to use each type of translator.</p>	<p>Students will need to know that:</p> <ul style="list-style-type: none"> • assemblers and compilers translate their input into machine code directly • interpreters do not generate machine code directly (they call appropriate machine code subroutines within their own code to carry out commands).

3.4.5 Systems architecture

Content	Additional information
<p>Explain the role and operation of main memory and the following major components of a central processing unit (CPU) within the Von Neumann architecture:</p> <ul style="list-style-type: none"> • arithmetic logic unit • control unit • clock • register • bus. 	<p>A bus is a collection of wires through which data/signals are transmitted from one component to another.</p> <p>Knowledge of specific registers is not required.</p> <p>Main memory will be considered to be any form of memory that is directly accessible by the CPU, except for cache and registers.</p>
<p>Explain the effect of the following on the performance of the CPU:</p> <ul style="list-style-type: none"> • clock speed • number of processor cores • cache size. 	
<p>Understand and explain the Fetch-Execute cycle.</p>	<p>The CPU continuously reads instructions stored in main memory and executes them as required:</p> <ul style="list-style-type: none"> • fetch: the next instruction is fetched to the CPU from main memory • decode: the instruction is decoded to work out what it is • execute: the instruction is executed (carried out). This may include reading/writing from/to main memory.

AQA GCSE Computer Science 8525 from 2020

<p>Understand the different types of memory within a computer:</p> <ul style="list-style-type: none"> • RAM • ROM • Cache • Register. <p>Know what the different types of memory are used for and why they are required.</p>	
<p>Understand the differences between main memory and secondary storage. Understand the differences between RAM and ROM.</p>	<p>Students should be able to explain the terms volatile and non-volatile. Secondary storage is considered to be any non-volatile storage mechanism not directly accessible by the CPU.</p>
<p>Understand why secondary storage is required.</p>	
<p>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.</p>	<p>Students should be aware that SSDs use electrical circuits to persistently store data but will not need to know the precise details such as use of NAND gates.</p>
<p>Explain the term cloud storage.</p>	<p>Students should understand that cloud storage uses magnetic and/or solid-state storage at a remote location.</p>
<p>Explain the advantages and disadvantages of cloud storage when compared to local storage.</p>	
<p>Understand the term embedded system and explain how an embedded system differs from a non-embedded system.</p>	<p>Students must be able to give examples of embedded and non-embedded systems.</p>

3.5 Fundamentals of computer networks

Content	Additional information
<p>Define what a computer network is.</p> <p>Discuss the advantages and disadvantages of computer networks.</p>	
<p>Describe the main types of computer network including:</p> <ul style="list-style-type: none"> • Personal Area Network (PAN) • Local Area Network (LAN) • Wide Area Network (WAN). 	<p>PAN - only Bluetooth needs to be considered.</p> <p>LAN - know that these usually cover relatively small geographical areas.</p> <p>LAN - know that these are often owned and controlled/managed by a single person or organisation.</p> <p>WAN - know that the Internet is the biggest example of a WAN.</p> <p>WAN - know that these usually cover a wide geographic area.</p> <p>WAN - know that these are often under collective or distributed ownership.</p>
<p>Understand that networks can be wired or wireless.</p> <p>Discuss the advantages and disadvantages of wireless networks as opposed to wired networks.</p>	<p>Students should know that wired networks can use different types of cable such as fibre and copper and when each would be appropriate.</p>
<p>Describe the following common LAN topologies:</p> <ul style="list-style-type: none"> • star • bus. 	<p>Students should be able to draw topology diagrams and describe the differences between the two topologies. They should also be able to select the most appropriate topology for a given scenario.</p>
<p>Define the term network protocol.</p>	
<p>Explain the purpose and use of common network protocols including:</p> <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • SMTP (Simple Mail Transfer Protocol) • IMAP (Internet Message Access Protocol). 	<p>Students should know what each protocol is used for (eg HTTPS provides an encrypted version of HTTP for more secure web transactions).</p> <p>Students should understand that Ethernet is a family of related protocols rather than a single protocol. They do not need to know the individual protocols that make up the Ethernet family.</p> <p>Students should understand that Wi-Fi is a family of related protocols rather than a single protocol. They do not need to know the individual protocols that make up the Wi-Fi family but they should know that Wi-Fi is a trademark and that the generic term for networks of this nature is WLAN.</p>
<p>Understand the need for, and importance of, network security.</p>	
<p>Explain the following methods of network security:</p>	<p>Students should be able to explain, using</p>

AQA GCSE Computer Science 8525 from 2020

<ul style="list-style-type: none"> • authentication • encryption • firewall • MAC address filtering. 	<p>examples, what each of these security methods is and when each could be used. Students should understand how these methods can work together to provide a greater level of security. The capabilities of firewalls have changed dramatically in recent years and will continue to do so. Students should be aware that a firewall is a network security device that monitors incoming and outgoing network traffic and decides whether to allow or block specific traffic based on a defined set of security rules. Students should understand that MAC address filtering allows devices to access, or be blocked from accessing a network based on their physical address embedded within the device's network adapter.</p>
<p>Describe the 4 layer TCP/IP model:</p> <ul style="list-style-type: none"> • application layer • transport layer • internet layer • link layer. <p>Understand that the HTTP, HTTPS, SMTP, IMAP and FTP protocols operate at the application layer.</p> <p>Understand that the TCP and UDP protocols operate at the transport layer.</p> <p>Understand that the IP protocol operates at the internet layer.</p>	<p>Students should be able to name the layers and describe their main function(s) in a networking environment.</p> <p>Application layer: this is where the network applications, such as web browsers or email programs, operate.</p> <p>Transport layer: this layer sets up the communication between the two hosts and they agree settings such as 'language' and size of packets.</p> <p>Internet layer: addresses and packages data for transmission. Routes the packets across the network.</p> <p>Link layer: this is where the network hardware such as the NIC (network interface card) is located. OS device drivers also sit here.</p> <p>Teachers should be aware that the link layer is sometimes referred to as the network access layer or network interface layer. However, students will not be expected to know these alternative layer names.</p>

3.6 Cyber security

3.6.1 Fundamentals of cyber security	
Content	Additional information
<p>Be able to define the term cyber security and be able to describe the main purposes of cyber security.</p> <p>Students should know that cyber security consists of the processes, practices and technologies designed to protect networks, computers, programs and data from attack, damage or unauthorised access.</p>	

3.6.2 Cyber security threats	
Content	Additional information
<p>Understand and be able to explain the following cyber security threats:</p> <ul style="list-style-type: none"> • social engineering techniques • malicious code (malware) • pharming • weak and default passwords • misconfigured access rights • removable media • unpatched and/or outdated software. 	<p>Pharming is a cyber-attack intended to redirect a website's traffic to a fake website.</p>
<p>Explain what penetration testing is and what it is used for.</p>	<p>Penetration testing is the process of attempting to gain access to resources without knowledge of usernames, passwords and other normal means of access.</p> <p>Students should understand that the aim of a white-box penetration test is to simulate a malicious insider who has knowledge of and possibly basic credentials for the target system.</p> <p>Students should understand that the aim of a black-box penetration test is to simulate an external hacking or cyber warfare attack where the attacker has no knowledge of any credentials for the target system.</p>

AQA GCSE Computer Science 8525 from 2020

3.6.2.1 Social engineering

Content	Additional information
<p>Define the term social engineering.</p> <p>Describe what social engineering is and how it can be protected against.</p> <p>Explain the following forms of social engineering:</p> <ul style="list-style-type: none">• blagging (pretexting)• phishing• shouldering (or shoulder surfing).	<p>Students should know that social engineering is the art of manipulating people so they give up confidential information.</p> <p>Blagging is the act of creating and using an invented scenario to engage a targeted victim in a manner that increases the chance the victim will divulge information or perform actions that would be unlikely in ordinary circumstances.</p> <p>Phishing is a technique of fraudulently obtaining private information, often using email or SMS.</p> <p>Shouldering is observing a person's private information over their shoulder eg cashpoint machine PIN numbers.</p>

3.6.2.2 Malicious code (malware)

Content	Additional information
<p>Define the term malware.</p> <p>Describe what malware is and how it can be protected against.</p> <p>Describe the following forms of malware:</p> <ul style="list-style-type: none">• computer virus• trojan• spyware.	<p>Malware is an umbrella term used to refer to a variety of forms of hostile or intrusive software.</p>

3.6.3 Methods to detect and prevent cyber security threats

Content	Additional information
<p>Understand and be able to explain the following security measures:</p> <ul style="list-style-type: none">• biometric measures (particularly for mobile devices)• password systems• CAPTCHA (or similar)• using email confirmations to confirm a user's identity• automatic software updates.	

3.7 Relational databases and structured query language (SQL)

3.7.1 Relational databases	
Content	Additional information
Explain the concept of a database.	
Explain the concept of a relational database.	
<p>Understand the following database concepts:</p> <ul style="list-style-type: none"> • table • record • field • primary key • foreign key. <p>Understand that the use of a relational database facilitates the elimination of data inconsistency and data redundancy.</p>	<p>Note that whilst the terms entity, attribute and entity identifier are more commonly used when an abstract model of a database is being considered, the terms given here will be used for both implementations of and abstract models of databases.</p>

3.7.2 Structured query language (SQL)	
Content	Additional information
<p>Be able to use SQL to retrieve data from a relational database, using the commands:</p> <ul style="list-style-type: none"> • SELECT • FROM • WHERE • ORDER BY...ASC DESC 	<p>Exam questions will require that data is extracted from no more than two tables for any one query.</p>
<p>Be able to use SQL to insert data into a relational database using the commands.</p> <p>INSERT INTO table_name (column1, column 2 ...) VALUES (value1, value2 ...)</p> <p>Be able to use SQL to edit and delete data in a database using the commands.</p> <p>UPDATE table name SET column1 = value1, column2 = value2 ... WHERE condition</p> <p>DELETE FROM table_name WHERE condition</p>	

3.8 Ethical, legal and environmental impacts of digital technology on wider society, including issues of privacy

Content	Additional information
<p>Explain the current ethical, legal and environmental impacts and risks of digital technology on society. Where data privacy issues arise, these should be considered.</p>	<p>Exam questions will be taken from the following areas:</p> <ul style="list-style-type: none">• cyber security• mobile technologies• wireless networking• cloud storage• hacking (unauthorised access to a computer system)• wearable technologies• computer-based implants• autonomous vehicles. <p>Students will be expected to understand and explain the general principles behind the issues rather than have detailed knowledge on specific issues.</p> <p>Students should be aware that ordinary citizens normally value their privacy and may not like it when governments or security services have too much access.</p> <p>Students should be aware that governments and security services often argue that they cannot keep their citizens safe from terrorism and other attacks unless they have access to private data.</p>