GCSE

eduqas
Part of WJEC

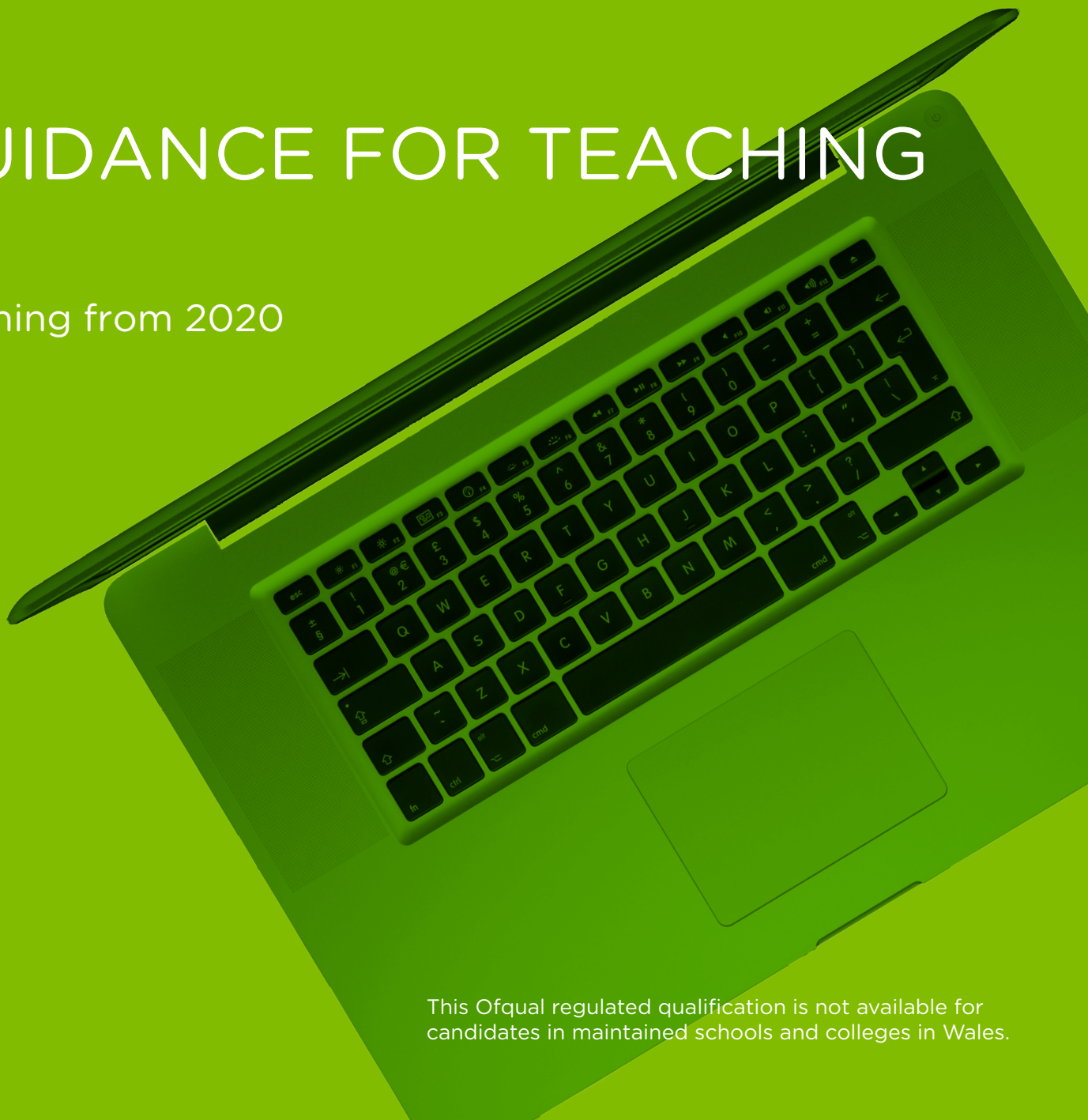
WJEC Eduqas GCSE in

COMPUTER SCIENCE

ACCREDITED BY OFQUAL

GUIDANCE FOR TEACHING

Teaching from 2020



wjec
cbac

This Ofqual regulated qualification is not available for candidates in maintained schools and colleges in Wales.

Introduction

The WJEC Eduqas GCSE Computer Science qualification, submitted to Ofqual for accreditation for first teaching from September 2020, is available to:

- all schools and colleges in England
- schools and colleges in independent regions such as Northern Ireland, Isle of Man and the Channel Islands
- independent schools in Wales.

It will be awarded for the first time in summer 2022, using grades 9 to 1.

Our GCSE Computer Science specification has two components. It is designed to allow learners to apply their knowledge of the theory of computer science and to link this seamlessly with programming and computational thinking.

The specification builds on the reputation WJEC has established with its current GCSE in Computer Science for clear, reliable assessment supported by straightforward, accessible guidance and administration. We have a proven track record of successfully providing assessment that caters for a wide range of ability and successfully motivates many learners towards further study in the subject.

Key features include:

- opportunities for flexible teaching approaches
- straightforward wording of questions
- accessibility of materials
- a range of question types
- opportunities for producing extended writing
- opportunities to link theoretical and practical concepts
- high-quality examination and resource materials.

The full set of requirements is outlined in the specification which can be accessed on the Eduqas website.

In addition to this Guidance, support is provided in the following ways:

- Sample Assessment Materials
- face-to-face and remote CPD events
- examiners' reports on each question paper
- free access to past question papers and mark schemes via the secure website
- direct access to the Subject Officer
- free online resources
- Exam Results Analysis
- Online Examination Review.

Aims of the Guidance for Teaching

The principal aims of the Guidance for Teaching are to offer support to teachers in their delivery of the new WJEC Eduqas GCSE in Computer Science specification and offer guidance as to the requirements of the qualification and the assessment process.

The Guidance is **not intended as a comprehensive reference**, but as support for teachers to develop stimulating and exciting courses tailored to the needs and skills of their own students in their particular institutions. The Guidance offers assistance to teachers with support from the Principal Examiners, who highlight areas of their components as an expansion of the information in the specification.

© WJEC CBAC Ltd.

Information for teachers transitioning from the 2016 Eduqas (9-1) GCSE Computer Science Specification

This section details topics that have changed from the 2016 Eduqas (9-1) GCSE Computer Science Specification. These changes fall into the following three categories:

1. **Topics that have been removed**
2. **Topics that include more detail**
3. **New topics**

Categories 1 and 2 above are not mutually exclusive; as a removed topic may have been replaced by a different topic that covers content originally defined under a different heading, but has been strengthened. For example IP and TCP (as separate protocols) and protocol stacks have been included but in their place TCP/IP and the 5 layer model have been removed.

1. Topics that have been removed

- All of Component 2 and 3
- RISC and CISC types of processors
- The requirement to know ring and bus networks in detail
- The difference between LAN and WAN network types
- The importance of connectivity
- Circuit switching
- 5 Layer network model
- The use of metadata in files
- User interface as a distinct part of operating systems

© WJEC CBAC Ltd.

2. Topics that include more detail

- Von Neumann architecture components explicitly stated with identified registers
- Primary storage more clearly defined
- Secondary storage more clearly defined
- Use of embedded systems changes to characteristics and purpose
- Appendix of Boolean rules and symbols defined in specification
- Hardware required to establish wired and wireless connectivity
- OSI 7 layer model replacing the previous 5 layer model
- Additional protocols including splitting TCP and IP
- Cybersecurity has been expanded and made more specific
- Representation of graphics and sound made more explicit
- Standardising on Base 2 representation for storage units e.g. 1 kilobyte = 1024 bytes
- Utility software now includes a list
- A new section in Component 1 about algorithms and constructs
- Software tools now has an explicit list
- Programming errors now has an explicit list
- Legislation now has an explicit list

3. New topics

- A new Component 2
- Importance of creating network standards
- Binary subtraction, two's complement and sign and magnitude representation
- Underflow

Guidance for Teaching

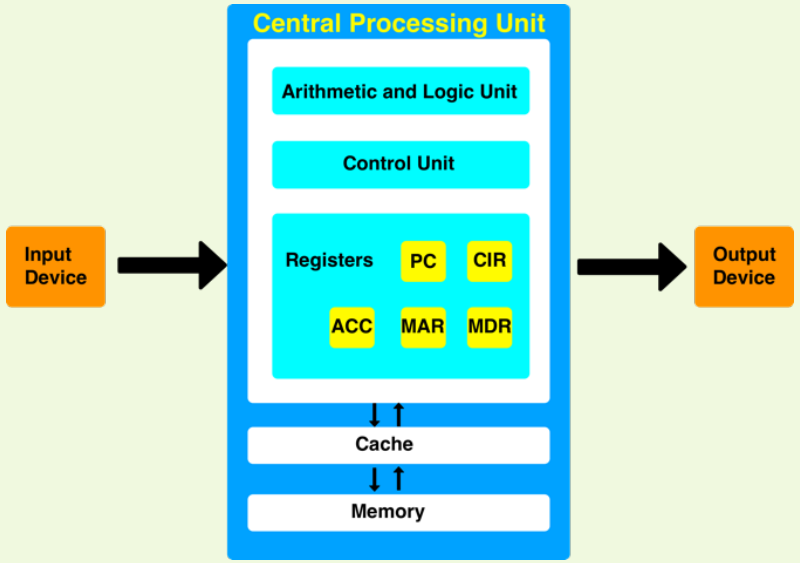
The following section is intended to give teachers guidance on the specification. It is designed to stimulate ideas for teaching and possible subject areas that may be explored to expand upon the content of the specification. It is not an exhaustive list and is not meant to be a substitute for sound classroom teaching.

Teachers should use this expansion in conjunction with class notes, exercises and textbooks or similar resources.

© WJEC CBAC Ltd.

1. Hardware

Content	Amplification	Teacher Guidance
	Candidates should be able to demonstrate and apply knowledge and understanding on the following:	

Central Processing Unit (CPU)	• the purpose of the CPU
	• the purpose of different components in the Von Neumann CPU architecture, including:
	  • control unit (CU)
	  • arithmetic and logic unit (ALU)
	  • main memory
	  • cache
	  • registers
	• program counter (PC)
	• current instruction register (CIR)
	• accumulator (ACC)
	• memory address register (MAR)
	• memory data register (MDR)
	• input and output devices.
		

© WJEC CBAC Ltd.

Content	Amplification	Teacher Guidance
	• the role of the components of the CPU in the fetch-decode-execute cycle.	Know that instructions have to be fetched from memory, decoded within the CPU and then executed. Registers: • Program Counter (PC) – a counter that keeps track of the memory address of which instruction is to be executed next. • Memory Address Register (MAR) – the address in main memory that is currently being read or written. • Current Instruction register (CIR) – a temporary holding area for the instruction that has just been fetched from memory. Components: • Control Unit (CU) / Clock – decodes the program instruction in the CIR, selecting machine resources and a particular arithmetic operation, and coordinates activation of those resources. • Arithmetic logic unit (ALU) – performs mathematical and logical operations.
	CPU performance in relation to: • cache size and levels of cache • clock speed • number of cores.	Cache size • More cache memory improves the performance as it can provide instructions and data to the CPU at a much faster rate than other system memory such as RAM. • More cache memory will allow more instructions that are repeatedly used by a CPU to be stored, and therefore increase the hit rate; increasing performance as a result. • L1 cache 2-64KB smallest and fastest. • L2 cache 256-512KB if instructions are not present in L1, L2 is searched, some latency due to increased size. • L3 cache 1-8MB, instructions are searched for in here if not present in L1 and L2. Largest and slowest cache but still faster than RAM.

© WJEC CBAC Ltd.

		Clock speed • The faster the clock speed, the faster the computer is able to run the fetch-decode-execute cycle and therefore process more instructions. • The faster the clock speed, the more power is generally required which creates greater requirements for heat dissipation and can place more strain on battery life. • Impact on temperature can be damaging. Number of cores • In a single-core CPU each instruction is processed one after the other, whereas in a dual-core CPU, two instructions may be processed at the same time. In theory, dual-core CPU should mean that the computer can process instructions twice as fast as a single-core CPU. • Performance may be affected where one core is waiting on the result of another and therefore cannot carry out any more instructions, leading to the performance being no better than a single core processor. • However multiple cores increase processor cost.

Primary storage	• the use and characteristics of primary storage, including:
	  • random access memory (RAM)
	  • read only memory (ROM)
	  • flash memory
	  • cache memory
	  • virtual memory.
		Understand that all memory stores information as bits and bytes. RAM is a volatile memory type, which stores programs and data currently in use.
		
		ROM is non-volatile memory type, which stores programs and data that does not change.
		
		Flash memory can be NAND memory. Flash memory is used for the permanent storage of data. This means that data is not lost when the power is switched off. However, the data stored in flash memory can be changed.
		
		Cache memory is volatile and can be used for the temporary storage of frequently accessed data and instructions. It consists of a small

© WJEC CBAC Ltd.

		number of store locations that can be accessed very quickly by the CPU; quicker than RAM. Virtual memory incorporates paging and swapping and is a memory management technique that gives the illusion of a very large main memory by moving data out to slower media e.g. disk when it is not being used. Understand and be able to give uses for these types of memory in different contexts.
Additional hardware components	the role of additional hardware, including: • integrated and dedicated GPUs • sound cards • motherboards.	Understand that additional hardware is required by computers to execute tasks. For example, the GPU is a specialised electronic circuit designed to rapidly manipulate and alter memory. GPUs efficiently manipulate computer graphics and carry out image processing.
Secondary storage	the functional characteristics of contemporary secondary storage technologies, including: • optical • magnetic • solid state • cloud storage. the suitability and characteristics of contemporary secondary storage media in terms of: • capacity • durability • portability • speed • cost.	Magnetic storage – disk/tape Optical storage – CD/DVD/Blu-ray Solid state storage – Flash/SD Relate secondary storage types to a situation or use. Suitability relates to overall suitability to a task and can be based on any of the below: Durability can relate to resistance to damage or longevity. Portability relates to removability and size. Speed relates to access time. For example: Magnetic storage: • Media: hard disk drives • Suitability: Backing up a home computer system • Capacity: 500 MB – 4 TB

© WJEC CBAC Ltd.

		• Durability: ★ • Portability: Yes • Speed of access: ★★★ Optical storage: • Media: CD/DVD/Blu-ray • Suitability: Storing multimedia files • Capacity: 650 MB (CD), 9GB (DVD), 50 GB (Blu-ray) • Durability: ★★★ • Portability: Yes • Speed of access: ★★ Solid state: • Media: Flash memory drives • Suitability: Moving relatively small files from work to home • Capacity: 2 GB – 4TB • Durability: ★★★★ • Portability: Yes Speed of access: ★★★★
Embedded systems	• the characteristics and purpose of embedded systems • example uses of embedded systems.	Understand that embedded systems exist in many devices in the home and outside and that these are small, dedicated computers. Embedded systems tend to be stored on ROM and are permanent (except in some cases where firmware is held on Flash memory and can be changed). Examples could include washing machines (the embedded system contains the washing programs), remote garage door entry systems, and engine sensors.

© WJEC CBAC Ltd.

Use an old PC – This will take place over a number of lessons

Architecture: Sometimes the scale of the CPU and internal components can be difficult to grasp. If you can obtain a PC which is not currently in use (most schools have to retain these until registered as discarded) it can be interesting to pull one apart. The key points here are to show that the CPU is a small integrated system and that RAM is separate. (Pull off the fan and heat sink, scrape off the thermal paste and pop the CPU out – then point out that the CPU is the tiny chip in the middle of the assembly.

Discuss that this tiny chip contains the CU, the ALU and the registers. Reinforce that the chip contains its own memory (registers and cache) but that the RAM is separate from the chip and this stores active programs while they are being run. This is an opportunity to explain the concepts of the CU, ALU and the registers/RAM.

Bits and Bytes: You can explain that RAM holds data stored as bits and bytes. Explain the concept of bits as 1 and 0 (on and off) as this is all essentially a computer can 'understand'. Explain that different combinations of 1 and 0 can represent an infinite range of data and that data is stored as bytes, kilobytes, megabytes etc.

Secondary Storage: There are opportunities here to discuss secondary storage drives whilst dismantling the PC and to see how these connect to the motherboard.

Discuss the suitability of different storage types in terms of capacity, portability, durability and speed of access.

It could be an idea to have a 'storage race' to discuss speed. Compare the transfer time of a large (>2GB) file onto magnetic HDD vs Optical vs Flash memory. The properties of the storage devices can be looked at here to look at capacities (e.g. the HDD will probably have a high GB or TB storage range whereas a DVD would only have 4.7GB, Flash memory would be variable but normally lie between the two).

Clock speed: You may also want to discuss clock speeds and that it is possible to overclock. This allows you to explain clock speed, and issues with overclocking (relate back to the heat sink and fan that you've already taken off the CPU).

Number of cores: Ask five students to volunteer to aid you in demonstrating the advantages of parallel processing. Explain to the class that a task such as sorting a deck of playing cards into order (A–K → Clubs, Diamonds, Hearts and Spades), can be completed much faster by four people (Quad-core processor) than by a single person (Single-core processor).

Give one deck to the student representing a single-core processor and give a quarter of the other deck to each student in the group of four representing a quad-core processor.

Both groups have to sort the deck as fast as possible.

Explain to the class that large problems can be divided into smaller ones, which are then solved concurrently and faster.

© WJEC CBAC Ltd.

Additional hardware components: The graphics card, sound card (may be integrated) and other can be shown in this exercise.

Embedded Systems: Point out the BIOS on the motherboard, explain that this is an example of ROM and it stores simple start-up sequences. Go on to explain that ROMs such as this can hold any program. Relate this to ROMs in washing machines etc. Explain that they may not always be ROMs and are sometimes NAND memory that can contain updateable firmware (e.g. car ECUs) that can be 'flashed' in order to update it.

© WJEC CBAC Ltd.

2. Logical operations

<table>
<tr><th>Topic</th><th>Amplification</th><th>Teacher Guidance</th></tr>
<tr><td></td><td colspan="2">Candidates should be able to demonstrate and apply knowledge and understanding on the following:</td></tr>
<tr><td>Logical operators</td><td>the use of logical operators in truth tables and to solve problems, including:
AND
OR
NOT
XOR</td><td>Understand the basic principles of the 4 logical operations, for example:
1 AND 0 = 0
1 OR 0 = 1
NOT 1 = 0 (or NOT 0 =1)
1 XOR 0 = 1
Use these in combination for example:
(1 AND 0) OR (1 AND 1) = 1
NOT (1 XOR 0) = 0
<table><tr><th>Condition</th><th>Symbol</th><th>Example</th></tr><tr><td>AND</td><td>.</td><td>$A.B$</td></tr><tr><td>OR</td><td>+</td><td>$A+B$</td></tr><tr><td>NOT</td><td>overline</td><td>$\overline{A}$</td></tr><tr><td>XOR</td><td>$\oplus$</td><td>$A \oplus B$</td></tr></table></td></tr>
<tr><td>Boolean logic</td><td>The simplification of Boolean expressions using Boolean identities and rules</td><td>Use the following Boolean identities and rules:
<table><tr><th>Rule</th><th>Boolean expression</th></tr><tr><td>Annulment</td><td>$A+1=1$
$A.0=0$</td></tr><tr><td>Identity</td><td>$A+0=A$
$A.1=A$</td></tr><tr><td>Commutative</td><td>$A+B=B+A$
$A.B=B.A$</td></tr><tr><td>Complement</td><td>$A+\overline{A}=1$
$A.\overline{A}=0$</td></tr><tr><td>Idempotent</td><td>$A+A=A$
$A.A=A$</td></tr><tr><td>Distribution</td><td>$A.(B+C)=A.B+A.C$
$A+(B.C)=(A+B).(A+C)$</td></tr><tr><td>Absorption</td><td>$A+(A.B)=A$
$A.(A+B)=A$</td></tr><tr><td>Association</td><td>$A+(B+C)=(A+B)+C=A+B+C$
$A.(B.C)=A.B.C$</td></tr></table></td></tr>
</table>

© WJEC CBAC Ltd.

Logical operations - Operation chains:

First point out the difference between AND and OR in terms of a meal

'Do you want a sandwich **and** a drink?' differs from 'Do you want a sandwich **or** a drink?'

Once you have established that AND and OR are different, explain that:

1 AND 0 = 0, 0 AND 1 = 0, 1 AND 1 = 1, 0 AND 0 = 0 (explain that both must be 1 for the result to be 1)

1 OR 0 = 1, 0 OR 1 = 1, 1 OR 1 = 1, 0 OR 0 = 0 (explain that if there is one or more 1, then the result will be 1)

Next explain that XOR is a different operation where 'one or the other but never both' applies and that:

1 XOR 0 = 1, 0 XOR 1 = 1, 1 XOR 1 = 0, 0 XOR 0 = 0 (explain only one 1 will give a result of 1)

Finally explain that NOT 'reverses' a result

NOT 1 = 0, NOT 0 = 1

Now try an operation chain with the class. Give a sequence of logical operations, and call them out to the class/write them on the board/show on a presentation. For example start with this chain, do not give the answers A–D:

1 OR 0 = **AnswerA** (1)

AnswerA AND 0 = **AnswerB** (0)

AnswerB XOR 0 = **AnswerC** (1)

NOT **AnswerC** = **AnswerD** (0)

Finally, end with a flourish (this will draw on BODMAS, which you may have to teach if they haven't been taught it in mathematics yet).

NOT (**AnswerD** XOR 1) = **AnswerE** (1)

You can now ask the students about the result they have (this chain should have 1 as its final result), and work through the problem. Some will have got it wrong, but working through will help their understanding. Repeat with more chains in class, or as independent learning tasks.

Truth tables will be the next stage once students have an appreciation of simple operations. Explain that truth tables are shorthand for the work they have already done.

© WJEC CBAC Ltd.

Topic	Amplification	Teacher Guidance
	Candidates should be able to demonstrate and apply knowledge and understanding on the following:	
Networks	the characteristics of networks, and their advantages and disadvantages	Data and resource sharing Security Globalisation Backups Archiving Disaster recovery Cost Maintenance
	common network topologies, including star and mesh, and their advantages and disadvantages over ring and bus topologies	Understand the configuration and advantages and disadvantages of: Star – Where one or more central switch, hub or computer acts as a central conduit to transmit messages. Very reliable and high data transfer speeds are possible (fewer collisions) and easy to identify faults. Can be expensive to set up as switches and cabling expensive, if main switch fails, the network fails and bottlenecking can occur if too much data is passing through the central switch. Mesh – each node relays data for the network. All mesh nodes cooperate in the distribution of data in the network. This is very reliable, and a network can 'self-heal' by reconfiguring itself around broken paths. A network is complex, expensive and difficult to set up. A large part of the network may be redundant.
	the hardware required to establish wired and wireless connectivity, including: • routers • hubs • switches • bridges • wireless access points (WAPs) • network interface cards (NICs)	Specialist hardware is used to construct networks, such as: Switches • A switch analyses each packet of data and sends it to the computer it was intended for. Hubs • A hub copies all packets of data to all devices on the network. Routers • A router stores the addresses of computers on the network and transfers data between devices.

© WJEC CBAC Ltd.

		Gateways • A gateway joins together two networks that use different base protocols, e.g. links a LAN to WAN. Bridge • A bridge joins together two networks that use the same base protocols, e.g. links LAN to LAN. Wireless Access Points • A device that allows other WiFi devices to connect to a network. Network Interface Card • A computer hardware component that allows a device to connect to a network (either wired or wireless).
	the importance of creating networking standards	Interoperability amongst products and services from different vendors and across countries. This includes security as well as hardware, protocols, software and transportation of data.
	the purpose of each layer in the 7-layer Open Systems Interconnection model (OSI model)	The 7 layers of the OSI model are: 7. Application layer 6. Presentation layer 5. Session layer 4. Transport layer 3. Network layer 2. Data link layer 1. Physical layer

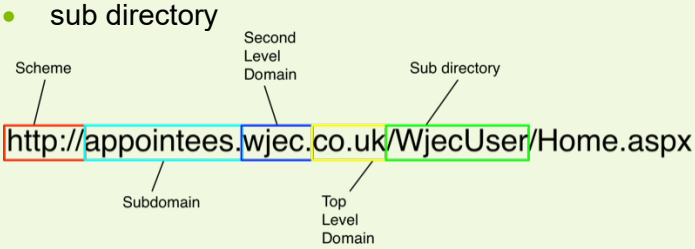
© WJEC CBAC Ltd.

OSI Model				
Layer			**Protocol Data Unit (PDU)**	**Function**
Host Layers	7	Application	Data	High-level APIs, including resource sharing, remote file access e.g. HTTP
	6	Presentation		Translation of data between a networking service and an application. This would include character encoding, data compression and encryption. e.g. ASCII or EBCDIC conversion
	5	Session		Managing communication sessions, so the exchange of information as transmissions between 2 nodes. E.g. full and half duplex as well as simplex operation
	4	Transport	Segment, Datagram	Reliable transmission of data segments between points on a network including segmentation, acknowledgement and multiplexing. e.g. TCP and UDP
Media Layers	3	Network	Packet	Structuring and managing a multi-node network including addressing, routing and traffic control e.g. IP
	2	Data Link	Frame	Reliable transmission of data frames between two nodes connected by a physical layer. The IEEE 802 standard defines Media Access Control (MAC - how a device gains access to a medium) and Logical Link Control(LLC error checking and frames). These are detailed in subsets of 802.11 (WiFi) and 802.3 (Ethernet)
	1	Physical	Symbol	Transmission and reception of raw bit streams over a physical medium e.g. Bluetooth, Ethernet, USB

© WJEC CBAC Ltd.

	the use of contemporary networking protocols in the 7-layer OSI model, including: • HTTP • HTTPS • SMTP • FTP • TCP • IP • Ethernet • WiFi (802.11)	Network protocols are vital to allow computers on networks to communicate. Without shared common protocols, computers would not be able to communicate. There are many types of protocol: Ethernet – wired (cable connection) protocol. Wi-Fi – wireless, two common standards are Bluetooth and 801.11. TCP– transmission control protocol– a means of putting datagrams together IP - internet protocol - a means of addressing and routing data packets across a network HTTP – hypertext transfer protocol – allows webpages to be shared across different computers and browsers. HTTPS – A secure variant of HTTP – it works together with another protocol, Secure Sockets Layer (SSL), to transport data securely. SMTP – simple mail transfer protocol -mail servers use SMTP to send and receive mail messages, mail applications typically use SMTP only for sending messages to a mail server. FTP – file transfer protocol – a means of sending files across the internet between one or more points.
	the typical contents of a TCP/IP packet and packet switching	The typical contents of a TCP/IP packet are: • The source address • The destination address • Information which enables the data to be reassembled into its original form • Other tracking information • The data itself • A checksum that checks that the data has not been corrupted. Packet switching is a method of grouping data into fixed or variable length packets to be sent over a network using a header containing destination information and a payload.

© WJEC CBAC Ltd.

	methods of routing traffic on a network and calculate routing costs	Understand that during routing, networks will search for the shortest path and the fastest nodes to transfer data. Together the path between nodes and the speed of the nodes are assessed by the device transmitting data. Computers will look for the route with the lowest *cost* (that is the shortest path and fastest nodes) and transmit data via this lowest cost route.
Internet	the purpose of Domain Name System (DNS) servers and how they work	Understand how DNS servers are used to resolve web addresses into IP addresses. Understand that IP addresses are difficult to remember compared to web addresses and that this is why DNS is vital to the ease of use of web addresses.
	the structure of URLs and the role and function of a web browser	A URL consists of 5 parts: • the scheme • subdomain • top level domain • second level domain • sub directory 
Cybersecurity	the characteristics of different threats to computer systems, including: • malware • phishing • social engineering • brute force attacks • denial of service attacks • data interception and theft • SQL injection	Malware • Short for malicious software, malware is a broad-spectrum term used to describe software used to disrupt computer operation. Phishing • Fraud where an Internet user is tricked (an email message is common) into revealing personal or confidential information e.g. bank details which can then be used fraudulently. Social Engineering • Psychological manipulation of people into revealing personal or confidential information e.g. bank details which can then be used fraudulently. Brute force attacks • Repeatedly trying combinations of characters in usernames and passwords in order to gain access to a computer system. Denial of service attacks • Prevention of access to systems usually by repeatedly sending huge amounts of messages asking a network or server to authenticate a request that has no valid return address.

© WJEC CBAC Ltd.

		Data Interception and theft • Unauthorised taking or interception of computer-based information SQL injection • Exploits poor security by adding SQL statements into data entry fields that destroy or copy data.
	different ways of protecting against threats during system design, creation, testing and use, including: • penetration testing • network forensics • anti-malware software • firewalls • user access levels • passwords • double authentication • encryption	Penetration testing • Attempting to penetrate a system's security layers in order to demonstrate security risks. Network Forensics • Monitoring and analysis of network traffic to detect intrusion. Anti-malware software • Anti-virus, spyware, ransomware and malicious website detection software Firewalls • Network security system that monitors incoming and outgoing traffic based on pre-determined rules User access levels • Assigning different levels of system access to users depending on their role. Passwords • A hidden keyword that may have to follow certain rules which validates a user. Double authentication • Prevents access unless two or more pieces of evidence are successfully given to the system e.g. at an ATM the card and the PIN. Encryption • Encoding data either symmetrically or asymmetrically in order to prevent access to the information.

© WJEC CBAC Ltd.

Topic	Amplification	Teacher Guidance
	Candidates should be able to demonstrate and apply knowledge and understanding on the following:	
Data types	data types, including: • integer • real • character • string • Boolean	Integer (whole number) 7, 0, 15, -5 Real (number with, or without, fraction) 7.2, 8.9, -6.8, 12.0 Character (single character) a, @, #, 8, Q String (one or more character) Hello, abc, b, Y Boolean (true/false)
Representation of numbers	• use and convert between denary, binary and hexadecimal counting systems	Direct conversion or using an intermediate base are both acceptable
	• the representation of positive and negative integers in a fixed-length store using both two's complement, and sign and magnitude representation	Signed binary numbers use the Most Significant Bit (MSB) to display a range of either positive or negative numbers In sign and magnitude: The MSB is the sign 0 for positive, 1 for negative. e.g. 0011 0101 represents +53 whereas 1011 0101 represents -53 as the sign bit has changed. Two's complement: The MSB is the sign 0 for positive, 1 for negative. e.g. 0011 0101 represents +53 whereas by flipping the bits and adding 1 1100 1011 represents -53
	• binary addition and subtraction techniques	Binary subtraction works by taking the Two's complement of the second number and adding it to the first e.g. 0011 0101 (53) - 0001 0101 (21) => 0011 0101 + 1110 1011 => 1 0010 0000 the ninth bit overflows and is discarded => 0010 0000 (32)

© WJEC CBAC Ltd.

	• the concept of overflow and underflow	The above is an example of overflow where an addition takes place in 8 bits and requires a ninth bit in which case that bit is discarded. Underflow would occur when the number resulting from a calculation is too small to be represented e.g. in an 8 bit representation an arithmetic shift right on 0000 0001 would result in a number that can't be represented.
	• arithmetic shift functions and their effect	<u>Left shift</u> e.g. for the number 001100_2 (12_{10}) a shift one place to the left would give $011000_2 = 24_{10}$ Which is equivalent to multiplying by 2 a shift two places to the left would give $110000_2 = 48_{10}$ Which is equivalent to multiplying by 4 <u>Right shift</u> e.g. for the number 001100_2 (12_{10}) a shift one place to the left would give $000110_2 = 6_{10}$ Which is equivalent to dividing by 2 a shift two places to the left would give $000011_2 = 3_{10}$ Which is equivalent to dividing by 4

Representation of characters	• the digital storage of characters
	• standardised character sets, including:
	  • Unicode
	  • American Standard Code for Information Interchange (ASCII)
		Understand that characters are represented by binary numbers e.g. in ASCII
		
		01100001 represents **a**
		
		01000001 represents **A**
		
		(There is no need to remember the number that represents a character).
		
		Understand that standardised character sets allow for data interchange between different programs.
		
		Understand that ASCII and Unicode are two such standards.
		
		Unicode can represent more characters than ASCII, but is more memory-hungry.

© WJEC CBAC Ltd.

<table>
<tr>
<td>Representation of graphics</td>
<td>

the digital storage of graphics
the effect of different resolutions and colour depths on the quality of graphics
how to calculate the storage requirements for graphics using different dimensions and colour depths

</td>
<td>
Understand that both raster and vector graphics can be stored on a computer.

Understand that raster graphics are dot matrix data structures representing a grid of pixels and cannot scale up without loss of apparent quality. They tend to be large in terms of the memory required to store them.

A bitmap image is a type of raster image and is composed of many tiny parts, called pixels, which are often many different colours. It is possible to edit each individual pixel.

Vector graphics use geometrically primitive objects (geometric primitives) such as points, lines, curves, and shapes or polygons which are based on mathematical expressions to represent images. Vector graphics can be scaled up without loss of apparent quality. They are smaller than bitmap graphics in terms of the memory required to store them.

The storage requirement will depend on the number of bits per pixel (bpp), the number of rows and the number of columns.

<table>
<tr><th>Bits per pixel</th><th>Number of colours</th></tr>
<tr><td>1 bpp</td><td>2</td></tr>
<tr><td>2 bpp</td><td>4</td></tr>
<tr><td>3 bpp</td><td>8</td></tr>
<tr><td>4 bpp</td><td>16</td></tr>
<tr><td>5 bpp</td><td>32</td></tr>
<tr><td>6 bpp</td><td>64</td></tr>
<tr><td>7 bpp</td><td>128</td></tr>
<tr><td>8 bpp</td><td>256</td></tr>
<tr><td>10 bpp</td><td>1024</td></tr>
<tr><td>16 bpp</td><td>65536</td></tr>
<tr><td>24 bpp</td><td>16777216 (16.7 million)</td></tr>
<tr><td>32 bpp</td><td>4294967296 (4294 million)</td></tr>
</table>

The size of an image = rows * cols * bpp

In a colour image of 640 x 480 at 24bpp the size of the image will be:

640 x 480 x 24 = 7372800 bits, or 921600 bytes or 900 kilobytes
</td>
</tr>
</table>

© WJEC CBAC Ltd.

<table>
<tr>
<td>Representation of sound</td>
<td>

the digital storage of sound
the effect of different sampling rates on the quality of sound
how to calculate the storage requirements for sound using different sampling rates

</td>
<td>
That sound is stored as a digital representation. The digital representation is achieved by sampling (signal processing).

The sample quality can be affected by the sample rate and sample frequency. The higher the sample rate and frequency, the larger the resultant sample.

The size of the storage requirement will depend on the sample rate, the sample resolution and the length of the sound.

File size = sample rate x sample resolution x length of sound

or

File size = bit rate x length of sound

e.g.
Sample rate = 8KHz
Sample resolution = 16 bit
Length of sound = 30 seconds

=> 8000 x 16 x 30 = 3840000 bits
=> 480000 bytes
=> 468.75KB
</td>
</tr>
<tr>
<td>Storage requirements</td>
<td>

the relationship between data storage units using the Base 2 prefixes, including:

bit
nibble
byte
kilobyte (i.e. 1,024 bytes)
megabyte (i.e. 1,024 kilobytes)
gigabyte (i.e. 1,024 megabytes)
terabyte (i.e. 1,024 gigabytes)
petabyte (i.e. 1,024 terabytes)

the calculation of data capacity requirements

</td>
<td>
Bit as 1 or 0

Nybble as 4 bit block (and usefulness in conversion from hex-binary) e.g. $1101_2 \rightarrow$ C

Byte as 8 bits 10110110_2

Understand that prefix multipliers work from bytes as 2^3 increments.

<table>
<tr><th></th><th>Symbol</th><th>Value</th></tr>
<tr><td>Byte</td><td>B</td><td>8 bits</td></tr>
<tr><td>Kilobyte</td><td>KB</td><td>1024 bytes</td></tr>
<tr><td>Megabyte</td><td>MB</td><td>1024 Kb</td></tr>
<tr><td>Gigabyte</td><td>GB</td><td>1024 MB</td></tr>
<tr><td>Terabyte</td><td>TB</td><td>1024 GB</td></tr>
<tr><td>Petabyte</td><td>PB</td><td>1024 TB</td></tr>
<tr><td>Exabyte</td><td>EB</td><td>1024 PB</td></tr>
<tr><td>Zettabyte</td><td>ZB</td><td>1024 EB</td></tr>
<tr><td>Yottabyte</td><td>YB</td><td>1024 ZB</td></tr>
</table>
</td>
</tr>
</table>

© WJEC CBAC Ltd.

Compression	• the principles of data compression and the use of different compression types, including:
	  • lossy
	  • lossless
		Understand that compression reduces file size in memory terms.
		Lossy compression results in reduction of data quality following compression.
		Lossless compression results in no loss of data quality following compression.

	• the calculation of compression ratios	Calculate compression ratios based on before and after file sizes (e.g. a 10MB file compressed to a 2 MB file will have a file compression ratio of 5:1). Using ratios, calculate the size of files following compression. Using data (e.g. size of destination storage medium), calculate the compression ratio required to store data.

© WJEC CBAC Ltd.

Explaining bases

'Back to basics' – Many students will have forgotten that they learned to count in base 10, and will simply accept 'that's the way it is'.

You may want to start with the line:

'There are 10 types of people in the world, those who understand binary and those who don't' – leave it on the screen/board during the lesson. Tell the students that this is a fact, and they will hopefully be intrigued, or point out that it makes no sense.

Remind students that in primary school they will have written 'Hundreds, Tens and Units' above numbers that they were going to use in maths problems. Explain that by using this system, and 10 numbers (0-9) they had the building blocks to construct any number.

Now ask them what if they'd only had two numbers, 1 and 0. Could they then represent any number? Ask for the largest number that they could represent using just two digits. They will say '11' (unless they already understand binary).

Go on to explain that binary is a different base system. And rather than columns with 'thousands, hundreds, tens, units' the columns are divided …128, 64, 32, 16, 8, 4, 2, 1, and that with this system, any number can be represented.

Explain that binary as a system produces very long numbers that are difficult to handle and that although denary can be used as shorthand, a better system is to use base 16 (hexadecimal). Explain that we need 0-9, and some extra digits A-F, and that columns are in the order …256, 16, 1

Refer back to the *'There are 10 types of people in the world, those who understand binary and those who don't'*.

© WJEC CBAC Ltd.

Topic	Amplification	Teacher Guidance
	Candidates should be able to demonstrate and apply knowledge and understanding on the following:	

Data structures	• the design, interpretation and manipulation of data structures including:
	  • records
	  • one-dimensional arrays
	  • two-dimensional arrays
		Describe relates to being able to describe and identify a data structure (for instance drawing a representation of a one-dimensional array or a two-dimensional array).
		
		Design relates to being able to design an array or record that would be suitable for a particular purpose (e.g. to design an array suitable for storing sales data over a time period).
		
		Interpret relates to being able to utilise a data structure to select data or to convert a graphical representation of an array to a programmed form or vice versa.
		• Manipulate relates to adding, deleting and editing data within an array or record.

	• the selection, identification and justification of appropriate data structures for different situations	• This relates to selecting a correct data structure for a given situation (e.g. to design an array suitable for storing sales data over a time period) or to identify the most appropriate data structure from a selection of data structures, in each case being able to justify why the data type is suitable.
File design	• the design of files and records appropriate for a particular application	• As above, but in relation to files and records.

© WJEC CBAC Ltd.

Topic	Amplification	Teacher Guidance
	Candidates should be able to demonstrate and apply knowledge and understanding on the following:	
Operating systems	The purpose of the operating system	An operating system manages hardware and software for a computer providing common services for computer programs.
	how the operating system manages: • CPU • multi-tasking • interrupts • memory • backing store • peripherals • interface • security	The O/S manages resources, such as: • Manages peripherals such as input and output devices • Manages printing using spooling • Manages backing store • File compression • Disk de-fragmentation • Manages memory (RAM) • Manages processes • Manages security
Utility software	• the purpose and functionality of a range of utility software, including: • anti-virus • clipboard manager • system profiles • backup software • disk checkers • disk compression • disk defragmenters • disk formatters • disk partition editors • archivers • cryptographic utilities • data recovery • revision control • file managers	Explain the purpose and functionality of utility software.

© WJEC CBAC Ltd.

Topic	Amplification	Teacher Guidance
	Candidates should be able to demonstrate and apply knowledge and understanding on the following:	
Levels of computer language	• the characteristics and purpose of high-level and low-level languages	Low level programming languages do not resemble natural languages, such as English or Welsh. They are made up entirely of bit patterns (instructions or data) that can be executed directly by the CPU. A high level language is a programming language that allows code to be written which is similar to a natural human language, such as English. Some programmers prefer to use high level programming languages as they are easier to understand, learn and program as commands are more English-like and identifiers can be long and meaningful. High level programming languages also allow the use of powerful commands that perform quite complex tasks such as MsgBox in Visual Basic or the SORT clause in COBOL.
	• situations that require the use of a high-level or a low-level language	Low level language are useful when speed of execution is critical or when writing software which interfaces directly with the hardware, e.g. device drivers. High level languages are used when the execution speed is not critical, e.g. in common productivity applications, such as a word processor.

© WJEC CBAC Ltd.

Topic	Amplification	Teacher Guidance
	Candidates should be able to demonstrate and apply knowledge and understanding and analyse problems in computational terms on the following:	

Algorithms	• methods of defining algorithms, including pseudo-code and flowcharts *(see Appendix A)*
	• writing, correcting, testing and interpreting the function of algorithms that solve problems using:
	  • subroutines
	  • input
	  • output
	  • sequence, selection and iteration
	  • counts and rogue values
	  • local and global variables
	  • self-documenting identifiers
	  • string handling
	  • mathematical and logical operations *(see Appendix A)*
		See Appendix A of specification for detail on pseudo-code conventions and flowchart conventions used.
		
		Identify subroutines in algorithms and explain their function.
		
		Understand that subroutines are a sequence of instructions that perform a specific task.
		
		Use subroutines to solve given problems.
		
		Identify sequence, selection and iteration in algorithms and explain their function.
		
		Sequence: an action, or event, leads to the next ordered action in a predetermined order.
		
		Selection: In selection, a question is asked, and depending on the answer, the program takes one of two courses of action.
		
		Iteration: An iteration is a single pass through a set of instructions. Most programs contain loops of instructions that are executed over and over again. The computer repeatedly executes the loop, iterating through the loop.
		
		Iterations within a loop must eventually be terminated. Counts and rogue values can be used to terminate iteration loops.
		
		Counts count the number of times a process has occurred.
		
		Rogue values are values that fall outside the range or type of the 'normal' data that will be processed by a program.

© WJEC CBAC Ltd.

		In both cases a loop can be terminated if a count reaches a pre-set value or if the loop encounters a rogue value. Follow the flow of and dry run algorithms, making changes to the algorithms that will affect output or repair a 'broken' algorithm. Algorithms can be of sequence, selection and iteration types and can contain input, processing and output of values. Write algorithms to solve given problems. Algorithms can be of sequence, selection and iteration types and can contain input, processing and output of values.
Sorting	the characteristics of sort algorithms, including: • merge sort • bubble sort	Merge sort as a **divide and conquer** algorithm where: An unsorted list is divided into n sublists, each containing 1 element. The sublists are repeatedly merged to produce new sorted sublists until there is only 1 sorted sublist remaining. Bubble sort as a simple sorting algorithm where: The algorithm that repeatedly steps through the list to be sorted. The algorithm compares each pair of adjacent items and swaps them if they are in the wrong order.
Searching	the characteristics of search algorithms, including: • linear search • binary search	Use linear searches and understand that this is a simple search process where a list is searched until the required value is found. Use binary search algorithms and understand binary search as a **divide and conquer** algorithm where: The middle value in a list is inspected to see if it matches the search value.

© WJEC CBAC Ltd.

		If the middle value is greater than the search value, the upper half of the list is discarded. If it is less than the search value, the lower half is discarded. This process is repeated, with the list halving in size each time until the search value is found.
Validation and Verification	• the purpose of and techniques for data validation and verification • algorithms that validate and verify data	Validation: Range check, type check, format check, presence check, lookup table, check digit. Verification: Double keying, proofreading.

© WJEC CBAC Ltd.

Topic	Amplification	Teacher Guidance
	Candidates should be able to demonstrate and apply knowledge and understanding on the following:	
Software tools	Integrated Development Environment (IDE) tools, including: • editor • automatic formatting • automatic line numbering • automatic colour coding • linker • libraries • routines/subroutines • standard functions • loader • debugger • syntax error detection • error diagnostics • trace • break point • variable watch • memory inspector • statement completion • code optimisation • compilation and interpretation of code	Understand the ways in which the IDE assists in the development and debugging of programs

© WJEC CBAC Ltd.

Topic	Amplification	Teacher Guidance
	Candidates should be able to demonstrate and apply knowledge and understanding on the following:	
Translators	the purpose of the three common types of program translator: • compilers • interpreters • assemblers the appropriate use of the three common types of program translator the principal stages involved in the compilation process: • lexical analysis • syntax analysis • semantic analysis • code generation • code optimisation examples of programming errors: • logical • syntactical • run-time/execution • linking • rounding • truncation	An assembler is a program, which converts the low level assembly programming language into machine code. An interpreter is a program, which converts code one line at a time, into machine code and executes it. A compiler is a program that converts high level programs into machine code for execution at a later time (the entire program is converted, not one line at a time as with an interpreter). Lexical analysis: • Comments and unneeded spaces are removed. • Keywords, constants and identifiers are replaced by 'tokens'. Symbol table construction: • A symbol table is created which holds the addresses of variables, labels and subroutines. Syntax analysis: • Tokens are checked to see if they match the spelling and grammar expected, using standard language definitions. This is done by parsing each token to determine if it uses the correct syntax for the programming language • If syntax errors are found, error messages are produced. Semantic analysis: • Variables are checked to ensure that they are of the correct data type, e.g. real values are not being assigned to integers. • Operations are checked to ensure that they are legal for the type of variable being used, e.g. you

© WJEC CBAC Ltd.

		would not try to store the result of a division operation as an integer. Code generation: • Machine code is generated. Code optimisation: • Variables are checked to ensure that they have been properly declared and used. • Code optimisation may be employed to make it more efficient/faster/less resource intense.

© WJEC CBAC Ltd.

Topic	Amplification	Teacher Guidance
	Candidates should be able to demonstrate and apply knowledge and understanding on the following:	
Impacts	the ethical, legal, cultural, environmental and privacy issues linked to the use of computer systems	Digital divide. Working conditions. Environmental impact (links to environmental issues). The implications of privacy of private data. Black hat/white hat/grey hat hacking. The growing importance of cybersecurity in light of increasing frequency of attacks.
Legislation	the impact of relevant current legislation on Computer Science, including: • the General Data Protection Regulation (GDPR) and Data Protection Act 2018 • Computer Misuse Act 1990 • Copyright Designs and Patents Act 1988 • Creative Commons Licensing • Regulation of Investigatory Powers Act 2000 • Telecommunications Regulations Act 2000 • Freedom of Information Act 2000	An understanding of the basic principles, the intent of the legislation and the penalties for breaking the law.
Professional standards	the importance of conforming to professional standards, including formal and informal codes of ethical behaviour and conduct	A formal code would comprise an acceptable use policy or internet access policy. Any policy that is formally written and signed up to is considered a formal code. An informal code comprises expectations, customs, habits, personal integrity.

© WJEC CBAC Ltd.

Topic	Amplification	Teacher Guidance
	Working individually and with others the candidates should be able to apply knowledge and understanding and analyse problems in computational terms in the following:	
Investigation	use a systematic approach to problem solving including the use of decomposition and abstraction	Decomposition is the process of breaking a complex problem into smaller component parts. Abstraction is the process of removing unnecessary detail and simplifying. Generally, abstraction precedes decomposition.
	use abstraction effectively to model selected aspects of the external world in an algorithm or program	Using abstraction to remove unnecessary detail from a real-world situation and to model the simplified result in an algorithm or program.
	use abstraction effectively to appropriately structure programs into modular parts with clear, well-documented interfaces	Using abstraction to produce a well-structured (as simple and elegant as possible) program. Modular programs should be built of self-contained subroutines. Interfaces should be clear, unambiguous and, where documented, documentation should be as unambiguous as possible.
	analyse a set of requirements	A task will be released to centres each year which will form the context of the examination.
	program a solution that meets a set of requirements	Using the version of Python specified on the task (which will be the same version used in the examination) candidates should create their own solution to the problem to ensure they are familiar with the requirements of the component.

© WJEC CBAC Ltd.

Topic	Amplification	Teacher Guidance
Design	design and document the input and output facilities required to produce an effective user interface	Candidates should give due consideration to effective GUI usage.
	design and document all required data structures	Candidates should be able to design arrays, tuples, data dictionaries and lists and know which is most appropriate to use.
	using pseudo code, design and document the following routines: • validation and verification • data handling and processing • authentication	Candidates must be familiar with the use of pseudo code and will be penalised in the examination for using Python in its place. Candidates should know how to design and when it is appropriate to use various validation and verification techniques including: • range checks • presence checks • lookup table • check digit • length checks • format checks • double entry. Candidates should be able to design authentication techniques to ensure access to authorised users only.
Implementation	design, write, test and refine Python 3 code using the following skills: • create new and extend existing functions or methods • create new and edit existing objects • create new and extend existing Python 3 libraries • use variables (labels), operators, inputs, outputs and assignment • use a variety of data types, including integer, Boolean, real, character and string • use programming constructs to control the flow of a program, including:	The version of Python 3 will be specified on the initial task which will be carried forward on to the associated examination. Candidates will be supplied with a working unannotated program based on the initial task. They will be required to annotate, add and amend functionality to this program during the examination. Candidates should be familiar with basic Python library functions such as TkInter. Candidates should document code explaining its functionality.

© WJEC CBAC Ltd.

	• iteration (condition and counter controlled loops) • selection • sequence • use basic file handling, including: • open a file • read from a file into a variable • read from a file into an array • write to a file • write to a file from an array • close a file • create new and extend data structures and fixed length records to store data • use string manipulation • create new and extend lists, tuples and dictionaries (arrays) • use slicing • use mathematical and logical operations • create new and modify existing intuitive graphical user interfaces using the Python 3 built in libraries including: • text boxes • buttons • forms • create and modify data validation and verification routines • create and modify authentication management routines • create code for the solution that is self-documenting and uses meaningful identifiers • use a programming style that is consistent, including indentation and appropriate use of white space • use subroutines with well-defined interfaces	

© WJEC CBAC Ltd.

	• annotate code so that it is accessible to a competent third party • explain the solution or changes made to a solution.	
Testing	• design and document an effective testing strategy that will ensure that the final solution meets the requirements • describe how the outcomes of the testing process can be used to inform further development of the solution • design test data to include examples of typical, extreme and erroneous content • implement a test plan using typical, extreme and erroneous data where appropriate • present test outcomes with detailed and informed commentaries • demonstrate testing and refinement of code during development or in response to change in the requirements provided • explain the outcome of testing using given data or data that the candidate has designed.	Candidates should test their programs to ensure that they function correctly and produce the expected results. When inputting an integer between 1 and 32768: • Typical data could be 16184 • Extreme data could be 1 or 32768 • Erroneous data could be A or -3 or 3.14 Candidates should be able to justify their data choices.
Refinement	• demonstrate understanding of the requirements and any changes required by revised requirements • demonstrate understanding of changes by presenting evidence of developmental changes • demonstrate testing and refinement of code to improve efficiency	Candidates will be required to modify and test existing program code according to a new requirement. Candidates will be required to evaluate any changes they have made using the correct technical terminology.

© WJEC CBAC Ltd.

	• discuss the strengths and weaknesses of differing approaches to meeting the requirements • demonstrate an understanding of the technical terminology/concepts used during software development • evaluate the outcomes of the original given code • evaluate the outcomes of the changes made by additional requirements.	

© WJEC CBAC Ltd.