3 Subject content

This syllabus gives you the flexibility to design a course that will interest, challenge and engage your learners. Where appropriate you are responsible for selecting resources and examples to support your learners' study. These should be appropriate for the learners' age, cultural background and learning context as well as complying with your school policies and local legal requirements.

Computer Science is a practical subject and a range of practical exercises must be integral to the teaching of this qualification. It is important that learners develop their computational thinking skills by doing practical problem-solving and programming using appropriate resources. It is also expected that learners have the opportunity in class to write their own programs, as well as executing (running), testing and debugging them.

Any equipment and facilities should be adequate for learners to be able to satisfy the requirements of the syllabus. The hardware facilities needed will depend on the number of learners but must be sufficient for all learners to have enough time to practise their programming skills. Learners also need to have access to a system with direct-access file capability on backing store and hardcopy facilities.

Computer systems

1 Data representation

1.1 Number systems

Candidates should be able to:	Notes and guidance
1 Understand how and why computers use binary to represent all forms of data	• Any form of data needs to be converted to binary to be processed by a computer • Data is processed using logic gates and stored in registers
2 (a) Understand the denary, binary and hexadecimal number systems	• Denary is a base 10 system • Binary is a base 2 system • Hexadecimal is a base 16 system
(b) Convert between (i) positive denary and positive binary (ii) positive denary and positive hexadecimal (iii) positive hexadecimal and positive binary	• Values used will be integers only • Conversions in both directions, e.g. denary to binary or binary to denary • Maximum binary number length of 16-bit
3 Understand how and why hexadecimal is used as a beneficial method of data representation	• Areas within computer science that hexadecimal is used should be identified • Hexadecimal is easier for humans to understand than binary, as it is a shorter representation of the binary
4 (a) Add two positive 8-bit binary integers	
(b) Understand the concept of overflow and why it occurs in binary addition	• An overflow error will occur if the value is greater than 255 in an 8-bit register • A computer or a device has a predefined limit that it can represent or store, for example 16-bit • An overflow error occurs when a value outside this limit should be returned

1.1 Number systems continued

Candidates should be able to:	Notes and guidance
5 Perform a logical binary shift on a positive 8-bit binary integer and understand the effect this has on the positive binary integer	• Perform logical left shifts • Perform logical right shifts • Perform multiple shifts • Bits shifted from the end of the register are lost and zeros are shifted in at the opposite end of the register • The positive binary integer is multiplied or divided according to the shift performed • The most significant bit(s) or least significant bit(s) are lost
6 Use two's complement to represent positive and negative 8-bit binary integers	• Convert a positive binary or denary integer to a two's complement 8-bit integer and vice versa • Convert a negative binary or denary integer to a two's complement 8-bit integer and vice versa

1.2 Text, sound and images

Candidates should be able to:	Notes and guidance
1 Understand how and why a computer represents text and the use of character sets, including American standard code for information interchange (ASCII) and Unicode	• Text is converted to binary to be processed by a computer • Unicode allows for a greater range of characters and symbols than ASCII, including different languages and emojis • Unicode requires more bits per character than ASCII
2 Understand how and why a computer represents sound, including the effects of the sample rate and sample resolution	• A sound wave is sampled for sound to be converted to binary, which is processed by a computer • The sample rate is the number of samples taken in a second • The sample resolution is the number of bits per sample • The accuracy of the recording and the file size increases as the sample rate and resolution increase

1.2 Text, sound and images continued

Candidates should be able to:	Notes and guidance
3 Understand how and why a computer represents an image, including the effects of the resolution and colour depth	• An image is a series of pixels that are converted to binary, which is processed by a computer • The resolution is the number of pixels in the image • The colour depth is the number of bits used to represent each colour • The file size and quality of the image increases as the resolution and colour depth increase

1.3 Data storage and compression

Candidates should be able to:	Notes and guidance
1 Understand how data storage is measured	• Including: – bit – nibble – byte – kibibyte (KiB) – mebibyte (MiB) – gibibyte (GiB) – tebibyte (TiB) – pebibyte (PiB) – exbibyte (EiB) • The amount of the previous denomination present in the data storage size, e.g.: – 8 bits in a byte – 1024 mebibytes in a gibibyte
2 Calculate the file size of an image file and a sound file, using information given	• Answers must be given in the units specified in the question. Calculations must use the measurement of 1024 and not 1000 • Information given may include: – image resolution and colour depth – sound sample rate, resolution and length of track
3 Understand the purpose of and need for data compression	• Compression exists to reduce the size of the file • The impact of this is, e.g.: – less bandwidth required – less storage space required – shorter transmission time

1.3 Data storage and compression continued

Candidates should be able to:	Notes and guidance
4 Understand how files are compressed using lossy and lossless compression methods	• Lossless compression reduces the file size without permanent loss of data, e.g. run length encoding (RLE) • Lossy compression reduces the file size by permanently removing data, e.g. reducing resolution or colour depth, reducing sample rate or resolution

2 Data transmission

2.1 Types and methods of data transmission

Candidates should be able to:	Notes and guidance
1 (a) Understand that data is broken down into packets to be transmitted	
(b) Describe the structure of a packet	• A packet of data contains a – packet header – payload – trailer • The packet header includes the: – destination address – packet number – originator's address
(c) Describe the process of packet switching	• Data is broken down into packets • Each packet could take a different route • A router controls the route a packet takes • Packets may arrive out of order • Once the last packet has arrived, packets are reordered
2 (a) Describe how data is transmitted from one device to another using different methods of data transmission	• Including: – serial – parallel – simplex – half-duplex – full-duplex
(b) Explain the suitability of each method of data transmission, for a given scenario	• Including the advantages and disadvantages of each method
3 Understand the universal serial bus (USB) interface and explain how it is used to transmit data	• Including the benefits and drawbacks of the interface

2.2 Methods of error detection

Candidates should be able to:	Notes and guidance
1 Understand the need to check for errors after data transmission and how these errors can occur	• Errors can occur during data transmission due to interference, e.g. data loss, data gain and data change
2 Describe the processes involved in each of the following error detection methods for detecting errors in data after transmission: parity check (odd and even), checksum and echo check	• Including parity byte and parity block check
3 Describe how a check digit is used to detect errors in data entry and identify examples of when a check digit is used, including international standard book numbers (ISBN) and bar codes	
4 Describe how an automatic repeat query (ARQ) can be used to establish that data is received without error	• Including the use of: – positive/negative acknowledgements – timeout

2.3 Encryption

Candidates should be able to:	Notes and guidance
1 Understand the need for and purpose of encryption when transmitting data	
2 Understand how data is encrypted using symmetric and asymmetric encryption	• Asymmetric encryption includes the use of public and private keys

3 Hardware

3.1 Computer architecture

Candidates should be able to:	Notes and guidance
1 (a) Understand the role of the central processing unit (CPU) in a computer	• The CPU processes instructions and data that are input into the computer so that the result can be output
(b) Understand what is meant by a microprocessor	• A microprocessor is a type of integrated circuit on a single chip
2 (a) Understand the purpose of the components in a CPU, in a computer that has a Von Neumann architecture	• Including: – units: arithmetic logic unit (ALU) and control unit (CU) – registers: program counter (PC), memory address register (MAR), memory data register (MDR), current instruction register (CIR) and accumulator (ACC) – buses: address bus, data bus and control bus
(b) Describe the process of the fetch–decode–execute (FDE) cycle including the role of each component in the process	• How instructions and data are fetched from random access memory (RAM) into the CPU, how they are processed using each component and how they are then executed • Storing data and addresses into specific registers • Using buses to transmit data, addresses and signals • Using units to fetch, decode and execute data and instructions
3 Understand what is meant by a core, cache and clock in a CPU and explain how they can affect the performance of a CPU	• The number of cores, size of the cache and speed of the clock can affect the performance of a CPU
4 Understand the purpose and use of an instruction set for a CPU	• An instruction set is a list of all the commands that can be processed by a CPU and the commands are machine code
5 Describe the purpose and characteristics of an embedded system and identify devices in which they are commonly used	• An embedded system is used to perform a dedicated function, e.g. domestic appliances, cars, security systems, lighting systems or vending machines. This is different to a general purpose computer that is used to perform many different functions, e.g. a personal computer (PC) or a laptop

3.2 Input and output devices

Candidates should be able to:	Notes and guidance
1 Understand what is meant by an input device and why it is required	• Including: – barcode scanner – digital camera – keyboard – microphone – optical mouse – QR code scanner – touch screen (resistive, capacitive and infra-red) – two-dimensional (2D) and three-dimensional (3D) scanners
2 Understand what is meant by an output device and why it is required	• Including: – actuator – digital light processing (DLP) projector – inkjet printer – laser printer – light emitting diode (LED) screen – liquid crystal display (LCD) projector – liquid crystal display (LCD) screen – speaker – 3D printer
3 (a) Understand what is meant by a sensor and the purposes of sensors	• Limited to: – acoustic – accelerometer – flow – gas – humidity – infra-red – level – light – magnetic field – moisture – pH – pressure – proximity – temperature
(b) Identify the type of data captured by each sensor and understand when each sensor would be used, including selecting the most suitable sensor for a given context	

3.3 Data storage

Candidates should be able to:	Notes and guidance
1 Understand what is meant by primary storage	• Primary storage is directly accessed by the CPU • Including the role of: – random access memory (RAM) – read only memory (ROM) • Including why a computer needs both RAM and ROM, and the difference between them
2 Understand what is meant by secondary storage	• Secondary storage is not directly accessed by the CPU and is necessary for more permanent storage of data
3 Describe the operation of magnetic, optical and solid-state (flash memory) storage and give examples of each	• Magnetic storage uses platters which are divided into tracks and sectors. Data is read and written using electromagnets • Optical storage uses lasers to create and read pits and lands • Solid-state (flash memory) uses NAND or NOR technology. Transistors are used as control gates and floating gates
4 Describe what is meant by virtual memory, how it is created and used and why it is necessary	• Pages of data are transferred between RAM and virtual memory when needed
5 Understand what is meant by cloud storage	• Cloud storage can be accessed remotely in comparison to storing data locally • Physical servers and storage are needed to store data in cloud storage
6 Explain the advantages and disadvantages of storing data on the cloud in comparison to storing it locally	

3.4 Network hardware

Candidates should be able to:	Notes and guidance
1 Understand that a computer needs a network interface card (NIC) to access a network	
2 Understand what is meant by and the purpose of a media access control (MAC) address, including its structure	• A network interface card is given a MAC address at the point of manufacture • MAC addresses are usually written as hexadecimal • MAC addresses are created using the manufacturer code and the serial code
3 (a) Understand what is meant by and the purpose of an internet protocol (IP) address (b) Understand that there are different types of IP address	• An IP address is allocated by the network and they can be static or dynamic • Including the characteristics of and differences between IPv4 and IPv6
4 Describe the role of a router in a network	• A router sends data to a specific destination on a network • A router can assign IP addresses • A router can connect a local network to the internet

4 Software

4.1 Types of software and interrupts

Candidates should be able to:	Notes and guidance
1 Describe the difference between system software and application software and provide examples of each	• System software provides the services that the computer requires, including operating system and utility software • Application software provides the services that the user requires
2 Describe the role and basic functions of an operating system	• Including: – managing files – handling interrupts – providing an interface – managing peripherals and drivers – managing memory – managing multitasking – providing a platform for running applications – providing system security – managing user accounts

4.1 Types of software and interrupts continued

Candidates should be able to:	Notes and guidance
3 Understand how hardware, firmware and an operating system are required to run applications software	• Applications are run on the operating system • The operating system is run on the firmware • The bootloader (firmware) is run on the hardware
4 Describe the role and operation of interrupts	• Including: – how an interrupt is generated – how it is handled using an interrupt service routine – what happens as a result of the interrupts • Software interrupts include division by zero and two processes trying to access the same memory location • Hardware interrupts include pressing a key on the keyboard and moving the mouse

4.2 Types of programming language, translators and integrated development environments (IDEs)

Candidates should be able to:	Notes and guidance
1 Explain what is meant by a high-level language and a low-level language, including the advantages and disadvantages of each	• Advantages and disadvantages include: – ease of reading and writing code, e.g. low-level is hard to read – ease of debugging code – machine independence – direct manipulation of hardware
2 Understand that assembly language is a form of low-level language that uses mnemonics, and that an assembler is needed to translate an assembly language program into machine code	
3 Describe the operation of a compiler and an interpreter, including how high-level language is translated by each and how errors are reported	• A compiler translates the whole code at once before executing it, producing an executable file • An interpreter translates and executes the code line-by-line • A compiler provides an error report for the whole code if errors are detected • An interpreter stops execution when an error is found

4.2 Types of programming language, translators and integrated development environments (IDEs) continued

Candidates should be able to:	Notes and guidance
4 Explain the advantages and disadvantages of a compiler and an interpreter	• To include an understanding that an interpreter is mostly used when developing a program and a compiler is used to translate the final program
5 Explain the role of an IDE in writing program code and the common functions IDEs provide	• Including: – code editors – run-time environment – translators – error diagnostics – auto-completion – auto-correction – prettyprint

5 The internet and its uses

5.1 The internet and the world wide web

Candidates should be able to:	Notes and guidance
1 Understand the difference between the internet and the world wide web	• The internet is the infrastructure • The world wide web is the collection of websites and web pages accessed using the internet
2 Understand what is meant by a uniform resource locator (URL)	• A URL is a text-based address for a web page; it can contain the protocol, the domain name and the web page/file name
3 Describe the purpose and operation of hypertext transfer protocol (HTTP) and hypertext transfer protocol secure (HTTPS)	
4 Explain the purpose and functions of a web browser	• The main purpose of a web browser is to render hypertext markup language (HTML) and display web pages • Functions include: – storing bookmarks and favourites – recording user history – allowing use of multiple tabs – storing cookies – providing navigation tools – providing an address bar

5.1 The internet and the world wide web continued

Candidates should be able to:	Notes and guidance
5 Describe how web pages are located, retrieved and displayed on a device when a user enters a URL	• Including the role of: – the web browser – IP addresses – domain name server (DNS) – web server – HTML
6 Explain what is meant by cookies and how they are used, including session cookies and persistent cookies	• Cookies are used for functions, including: – saving personal details – tracking user preferences – holding items in an online shopping cart – storing login details

5.2 Digital currency

Candidates should be able to:	Notes and guidance
1 Understand the concept of a digital currency and how digital currencies are used	• A digital currency is one that only exists electronically
2 Understand the process of blockchain and how it is used to track digital currency transactions	• Blockchain, in its basic form, is a digital ledger, that is a time-stamped series of records that cannot be altered

5.3 Cyber security

Candidates should be able to:	Notes and guidance
1 Describe the processes involved in, and the aim of carrying out, a range of cyber security threats	• Including: – brute-force attack – data interception – distributed denial of service (DDoS) attack – hacking – malware (virus, worm, Trojan horse, spyware, adware, ransomware) – pharming – phishing – social engineering

5.3 Cyber security continued

Candidates should be able to:	Notes and guidance
2 Explain how a range of solutions are used to help keep data safe from security threats	• Including: – access levels – anti-malware including anti-virus and anti-spyware – authentication (username and password, biometrics, two-step verification) – automating software updates – checking the spelling and tone of communications – checking the URL attached to a link – firewalls – privacy settings – proxy-servers – secure socket layer (SSL) security protocol

6 Automated and emerging technologies

6.1 Automated systems

Candidates should be able to:	Notes and guidance
1 Describe how sensors, microprocessors and actuators can be used in collaboration to create automated systems	
2 Describe the advantages and disadvantages of an automated system used for a given scenario	• Including scenarios from: – industry – transport – agriculture – weather – gaming – lighting – science

6.2 Robotics

Candidates should be able to:	Notes and guidance
1 Understand what is meant by robotics	• Robotics is a branch of computer science that incorporates the design, construction and operation of robots • Examples include factory equipment, domestic robots and drones
2 Describe the characteristics of a robot	• Including: – a mechanical structure or framework – electrical components, such as sensors, microprocessors and actuators – programmable
3 Understand the roles that robots can perform and describe the advantages and disadvantages of their use	• Robots can be used in areas including: – industry – transport – agriculture – medicine – domestic – entertainment

6.3 Artificial intelligence

Candidates should be able to:	Notes and guidance
1 Understand what is meant by artificial intelligence (AI)	• AI is a branch of computer science dealing with the simulation of intelligent behaviours by computers
2 Describe the main characteristics of AI as the collection of data and the rules for using that data, the ability to reason, and can include the ability to learn and adapt	
3 Explain the basic operation and components of AI systems to simulate intelligent behaviour	• Limited to: – expert systems – machine learning • Expert systems have a knowledge base, a rule base, an inference engine and an interface • Machine learning is when a program has the ability to automatically adapt its own processes and/or data

Algorithms, programming and logic

See section 4 for the:

- standard flowchart symbols that must be used by students when drawing flowcharts
- logic gate symbols that must be used by students when drawing logic circuits
- format in which pseudocode will appear in examinations.

Students are advised to program solutions to a variety of different problems on a computer, using one of these high-level programming languages: Python, VB.NET or Java.

7 Algorithm design and problem-solving

Candidates should be able to:	Notes and guidance
1 Understand the program development life cycle, limited to: analysis, design, coding and testing	• Including identifying each stage and performing these tasks for each stage: – analysis: abstraction, decomposition of the problem, identification of the problem and requirements – design: decomposition, structure diagrams, flowcharts, pseudocode – coding: writing program code and iterative testing – testing: testing program code with the use of test data
2 (a) Understand that every computer system is made up of sub-systems, which are made up of further sub-systems	
(b) Understand how a problem can be decomposed into its component parts	• Including: – inputs – processes – outputs – storage
(c) Use different methods to design and construct a solution to a problem	• Including: – structure diagrams – flowcharts – pseudocode
3 Explain the purpose of a given algorithm	• Including: – stating the purpose of an algorithm – describing the processes involved in an algorithm

7 Algorithm design and problem-solving continued

Candidates should be able to:	Notes and guidance
4 Understand standard methods of solution	• Limited to: – linear search – bubble sort – totalling – counting – finding maximum, minimum and average values
5 (a) Understand the need for validation checks to be made on input data and the different types of validation check	• Including: – range check – length check – type check – presence check – format check – check digit – the purpose of each validation check and writing algorithms to implement each validation check
(b) Understand the need for verification checks to be made on input data and the different types of verification check	• Including: – visual check – double entry check – The purpose of each verification check
6 Suggest and apply suitable test data	• Limited to: – normal – abnormal – extreme – boundary • Extreme data is the largest/smallest acceptable value • Boundary data is the largest/smallest acceptable value and the corresponding smallest/largest rejected value
7 Complete a trace table to document a dry-run of an algorithm	• Including, at each step in an algorithm: – variables – outputs – user prompts
8 Identify errors in given algorithms and suggest ways of correcting these errors	

7 Algorithm design and problem-solving continued

9 Write and amend algorithms for given problems or scenarios, using: pseudocode, program code and flowcharts	• Precision is required when writing algorithms, e.g. `x > y` is acceptable but `x is greater than y` is not acceptable • See section 4 for flowchart symbols • See section 4 for pseudocode

8 Programming

8.1 Programming concepts

Candidates should be able to:	Notes and guidance
1 Declare and use variables and constants	
2 Understand and use the basic data types	• Including: – integer – real – char – string – Boolean
3 Understand and use input and output	
4 (a) Understand and use the concept of sequence	
(b) Understand and use the concept of selection	• Including: – `IF` statements – `CASE` statements
(c) Understand and use the concept of iteration	• Including: – count-controlled loops – pre-condition loops – post-condition loops
(d) Understand and use the concepts of totalling and counting	
(e) Understand and use the concept of string handling	• Including: – length – substring – upper – lower • The first character of the string can be position zero or one

8.1 Programming concepts continued

Candidates should be able to:	Notes and guidance
(f) Understand and use arithmetic, logical and Boolean operators	• Arithmetic, limited to: – `+` – `-` – `/` – `*` – `^` (raised to power of) – `MOD` – `DIV` • Logical, limited to: – `=` – `<` – `<=` – `>` – `>=` – `<>` (not equal to) • Boolean, limited to: – `AND` – `OR` – `NOT`
5 Understand and use nested statements	• Including nested selection and iteration • Candidates will **not** be required to write more than three levels of nested statements
6 (a) Understand what is meant by procedures, functions and parameters (b) Define and use procedures and functions, with or without parameters (c) Understand and use local and global variables	• Procedures and functions may have up to two parameters
7 Understand and use library routines	• Including: – MOD – DIV – ROUND – RANDOM

8.1 Programming concepts continued

Candidates should be able to:	Notes and guidance
8 Understand how to create a maintainable program	• Including appropriate use of: – meaningful identifiers – the commenting feature provided by the programming language – procedures and functions – relevant and appropriate commenting of syntax • Use meaningful identifiers for: – variables – constants – arrays – procedures and functions

8.2 Arrays

Candidates should be able to:	Notes and guidance
1 Declare and use one-dimensional (1D) and two-dimensional (2D) arrays	
2 Understand the use of arrays	• Including the use of variables as indexes in arrays • The first index can be zero or one
3 Write values into and read values from an array using iteration	• Including nested iteration

8.3 File handling

Candidates should be able to:	Notes and guidance
1 Understand the purpose of storing data in a file to be used by a program	
2 Open, close and use a file for reading and writing	• Including: – read and write single items of data – read and write a line of text

9 Databases

Candidates should be able to:	Notes and guidance
1 Define a single-table database from given data storage requirements	• Including: – fields – records – validation
2 Suggest suitable basic data types	• Including: – text/alphanumeric – character – Boolean – integer – real – date/time
3 Understand the purpose of a primary key and identify a suitable primary key for a given database table	
4 Read, understand and complete structured query language (SQL) scripts to query data stored in a single database table	• Limited to: – `SELECT` – `FROM` – `WHERE` – `ORDER BY` – `SUM` – `COUNT` • Identifying the output given by an SQL statement that will query the given contents of a database table

10 Boolean logic

Candidates should be able to:	Notes and guidance
1 Identify and use the standard symbols for logic gates	• See section 4 for logic gate symbols
2 Define and understand the functions of the logic gates	• Including: – NOT – AND – OR – NAND – NOR – XOR (EOR) – the binary output produced from all the possible binary inputs • NOT is a single input gate • All other gates are limited to two inputs
3 (a) Use logic gates to create given logic circuits from a: (i) problem statement (ii) logic expression (iii) truth table	• Circuits must be drawn for the statement given, without simplification • Logic circuits will be limited to a maximum of three inputs and one output
(b) Complete a truth table from a: (i) problem statement (ii) logic expression (iii) logic circuit	• An example truth table with three inputs, for completion:

A	B	C	Output
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

(c) Write a logic expression from a: (i) problem statement (ii) logic circuit (iii) truth table	