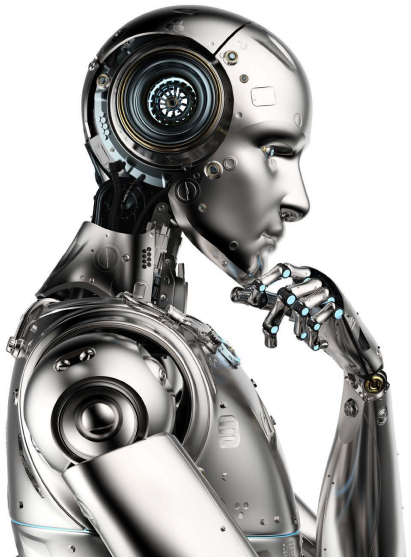
Cambridge Assessment International Education

Scheme of Work

Cambridge IGCSE™ / Cambridge IGCSE (9–1) Computer Science 0478 / 0984

Cambridge O Level Computer Science 2210

For examination from 2023



Version 2

Cambridge Pathway

Contents

Introduction

This scheme of work has been designed to support you in your teaching and lesson planning. Making full use of this scheme of work will help you to improve both your teaching and your learners' potential. It is important to have a scheme of work in place in order for you to guarantee that the syllabus is covered fully. You can choose what approach to take and you know the nature of your institution and the levels of ability of your learners. What follows is just one possible approach you could take and you should always check the syllabus for the content of your course.

Suggestions for independent study **(I)** and formative assessment **(F)** are also included. Opportunities for differentiation are indicated as **Extension activities**; there is the potential for differentiation by resource, grouping, expected level of outcome, and degree of support by teacher, throughout the scheme of work. Timings for activities and feedback are left to the judgment of the teacher, according to the level of the learners and size of the class. Length of time allocated to a task is another possible area for differentiation.

Guided learning hours

Guided learning hours give an indication of the amount of contact time you need to have with your learners to deliver a course. Our syllabuses are designed around 130 hours for Cambridge IGCSE and O Level courses. The number of hours may vary depending on local practice and your learners' previous experience of the subject. The table below give some guidance about how many hours we recommend you spend on each topic area.

Topic	Suggested teaching time (hours / % of the course)	Suggested teaching order
1 Data representation	It is recommended that this should take about 12 hours/ 9% of the course.	Topics 1–6 address elements that will be tested in Paper 1. Topics 7–10 address elements that will be tested in Paper 2. Units are best taught in sequence, as concepts developed in one unit will be applied in the following units. Schools may choose to teach Unit 1 first or both units in parallel, to balance theory with practical activity.
2 Data transmission	It is recommended that this should take about 8 hours/ 6% of the course.	
3 Hardware	It is recommended that this should take about 20 hours/ 16% of the course.	
4 Software	It is recommended that this should take about 8 hours/ 6% of the course.	
5 The internet and its uses	It is recommended that this should take about 8 hours/ 6% of the course.	
6 Automated and emerging technologies	It is recommended that this should take about 7 hours/ 6% of the course.	
7 Algorithm design and problem solving	It is recommended that this should take about 21 hours/ 16% of the course.	
8 Programming	It is recommended that this should take about 30 hours/ 23% of the course.	
9 Databases	It is recommended that this should take about 8 hours/ 6% of the course.	
10 Boolean logic	It is recommended that this should take about 8 hours/ 6% of the course.	

Resources

You can find the endorsed resources to support Cambridge IGCSE Computer Science and O Level Computer Science on the Published resources tab of the syllabus pages on our public website (IGCSE) (O Level)

Endorsed textbooks have been written to be closely aligned to the syllabus they support, and have been through a detailed quality assurance process. All textbooks endorsed by Cambridge International for this syllabus are the ideal resource to be used alongside this scheme of work as they cover each learning objective. Throughout this scheme of work we have referred to our Cambridge IGCSE Computer Science 0478 past papers. The 0478 syllabus runs parallel to the 0984 and the O Level Computer Science 2210 syllabus content and assessment criteria. Cambridge IGCSE, Cambridge IGCSE (9–1) and O Level Computer Science syllabuses are at the same level. In addition to reading the syllabus, teachers should refer to the updated specimen assessment materials.

Tools to support remote teaching and learning – Click here to find out about and explore the various online tools available for teachers and learners.

School Support Hub

The School Support Hub www.cambridgeinternational.org/support is a secure online resource bank and community forum for Cambridge teachers, where you can download specimen and past question papers, mark schemes and other teaching and learning resources. We also offer online and face-to-face training; details of forthcoming training opportunities are posted online. This scheme of work is available as PDF and an editable version in Microsoft Word format; both are available on the School Support Hub at www.cambridgeinternational.org/support. If you are unable to use Microsoft Word you can download Open Office free of charge from www.openoffice.org

Websites

This scheme of work includes website links providing direct access to internet resources. Cambridge Assessment International Education is not responsible for the accuracy or content of information contained in these sites. The inclusion of a link to an external website should not be understood to be an endorsement of that website or the site's owners (or their products/services).

The website pages referenced in this scheme of work were selected when the scheme of work was produced. Other aspects of the sites were not checked and only the particular resources are recommended.

How to get the most out of this scheme of work – integrating syllabus content, skills and teaching strategies

We have written this scheme of work for the Cambridge IGCSE and O Level Computer Science syllabuses and it provides some ideas and suggestions of how to cover the content of the syllabus. We have designed the following features to help guide you through your course.

Learning objectives help your learners by making it clear the knowledge they are trying to build. Pass these on to your learners by expressing them as 'We are learning to / about…'.

Suggested teaching activities give you lots of ideas about how you can present learners with new information without teacher talk or videos. Try more active methods which get your learners motivated and practising new skills.

Syllabus ref.	Learning objectives	Suggested teaching activities
10 Boolean logic	1 Identify and use standard symbols for logic gates. 2 Define and understand the functions of the logic gates. 3 Use of logic gates to create given logic circuits. Complete a truth table and write a logic expression.	Teacher introduces the concepts of OR and AND by careful use of English in logical statements; linking TRUE and FALSE to binary states (1 and 0). Teacher demonstrates OR gate and AND gate by use of electrical model, projector presentation of si similar. Teacher develops the concept of truth table for OR gate; learners develop truth table for AND Introduction, notes and activities at: https://cambridgegcsecomputing.org/boolean-logic#top Teacher demonstrates OR gate and AND gate by use of electrical model, projector presentation of si similar. Teacher develops the concept of truth table for OR gate; learners develop truth table for AND A useful presentation on binary logic is: om/watch?v=_76g8EM4DVU ces the standard symbols for both gates, see section 4 of the syllabus. propriate hardware or simulation software to develop understanding of the functions d XOR (EOR) logic gates. **(I)** All gates are limited to two inputs except NOT which is a ulator using standard symbols: http://logic.ly/demo/ gic gate simulator: www.softpedia.com/get/Others/Home-Ed t.org/ Learners can observe the output produced from all possible combinations of table. **(I)** **Extension activity:** Learners work out the simple logic circuits required to cr OR and NOT gates and test them. **(I)**

Independent study (I) gives your learners the opportunity to develop their own ideas and understanding without direct input from you.

Past papers, specimen papers and **mark schemes** are available for you to download at: **www.cambridgeinternational.org/support**

Using these resources with your learners allows you to check their progress and give them confidence and understanding.

Extension activities provide your more able learners with further challenge beyond the basic content of the course. Innovation and independent learning are the basis of these activities.

Past and specimen papers
Past/specimen papers and mark schemes are available to download at www.cambridgeinternational.org/support **(F)** March 2019 Paper 12, question 5 June 2019 Paper 11, question 2 June 2019 Paper 12, question 8

Formative assessment (F) is on-going assessment which informs you about the progress of your learners. Don't forget to leave time to review what your learners have learnt, you could try question and answer, tests, quizzes, 'mind maps', or 'concept maps'. These kinds of activities can be found in the scheme of work.

Data representation

Syllabus ref.	Learning objectives	Suggested teaching activities
1.1 Number systems	1 Understand how and why computers use binary to represent all forms of data.	Teacher presentation to introduce the concepts of binary notation, e.g. using an automatic binary counter; binary number cards available (with worksheets etc.). Binary numbers at Computer Studies Unplugged: http://csunplugged.org/binary-numbers
	2 Understand denary, binary and hexadecimal number systems. Convert positive integers between these number systems.	Learners convert positive denary integers to and from binary. Reinforce with a game such as the Cisco binary game. This provides formative assessment of understanding. Cisco binary game: www.games.penjee.com/binary-numbers-game/ Learners answer previous exam/textbook questions on binary representation. Teacher presentation on hexadecimal notation and its relationship to binary and denary notation. Hexadecimal counter – for example: www.mathsisfun.com/binary-decimal-hexadecimal-converter.html Demonstration of the conversion of binary and denary to hexadecimal and hexadecimal to denary and binary. Learners convert positive hexadecimal integers to and from binary and to and from denary. **(I)**
	3 Understand how and why hexadecimal is used as a beneficial method of data representation.	Class brainstorm to show understanding of the reasons for choosing hexadecimal to represent numbers, e.g. those stored in registers and main memory. Learners answer previous exam/textbook questions on hexadecimal representation. **(F)**
	4 Add two positive 8-bit binary integers. Understand the concept of overflow and why it occurs in binary addition.	Teacher presentation to introduce binary addition, limit this to 8-bit positive binary integers and include the concept of overflow. Binary addition including overflow: https://isaaccomputerscience.org/concepts/gcse_data_addition
	5 Perform logical binary shifts on positive 8-bit binary integers and understand the effect this has on the positive binary	Teacher presentation to introduce logical left and right binary shifts, limit this to 8-bit positive binary integers. Binary shifts: https://community.computingatschool.org.uk/resources/5086/single

Syllabus ref.	Learning objectives	Suggested teaching activities
	integer. 6 Use two's complement to represent positive and negative 8-bit binary integers	Teacher presentation to introduce two's complement to represent positive and negative numbers, limit this to 8-bit positive and negative binary integers. Learners use two's complement to represent negative binary integers.**(I)** Learners answer previous exam/textbook questions on two's complement.**(F)**
1.2 Text, Sound and Images	1 Understand how and why a computer represents text and the use of character sets, including ASCII and Unicode.	Provide leaners with a table example of a character set, such as ASCII. There are many examples available on the internet. Explain the need for character sets and how each character is assigned a code. Give learners a message in binary to decode using the character set. Then get learners to code a message for another class member using the character set. Comprehensive notes and exercises for ASCII: http://en.wikibooks.org/wiki/A-level_Computing/AQA/Problem_Solving,_Programming,_Data_Representation_and_Practical_Exercise/Fundamentals_of_Data_Representation/ASCII Comprehensive notes and exercises on Unicode: http://en.wikibooks.org/wiki/A-level_Computing/AQA/Problem_Solving,_Programming,_Data_Representation_and_Practical_Exercise/Fundamentals_of_Data_Representation/Unicode Learners answer previous exam/textbook questions on character sets
	2 Understand how and why a computer represents sound, including the effects of the sample rate and sample resolution.	Teacher presentation to introduce computer representation of sound. Provide learners with some examples: https://filestore.aqa.org.uk/resources/computing/AQA-8520-REP-SOUND-LP-PA.PDF https://www.trccompsci.online/mediawiki/index.php/Sound https://www.101computing.net/sampling-sound/ Notes and quiz: https://studyrocket.co.uk/revision/gcse-computer-science-aqa/written-assessment/representing-sound
	3 Understand how and why a computer represents an image, including the effects of the resolution and colour depth.	Teacher presentation to introduce computer representation of an image. Provide learners with some examples: http://teach-ict.com/2016/GCSE_Computing/OCR_J276/2_6_data_representation/images/miniweb/index.php Notes and quiz: https://studyrocket.co.uk/revision/gcse-computer-science-aqa/written-assessment/image-encoding Learners answer previous exam/textbook questions on text sound and images.**(F)**
1.3 Data storage and compression	1 Understand how data storage is measured.	Teacher presentation to explain the importance of compressing files that are transmitted via the internet: • describe the differences between lossy and lossless compression.

Syllabus ref.	Learning objectives	Suggested teaching activities
	2 Calculate file sizes for image and sound files. 3 Understand the purpose of and need for data compression. 4 Understand how files are compressed using lossy and lossless file compression methods.	Lossy and lossless compression notes: http://computer.howstuffworks.com/file-compression3.htm Learners complete a multiple-choice quiz and answer previous exam/textbook questions on data storage and compression. **(F)**

Past and specimen papers

Past/specimen papers and mark schemes are available to download at www.cambridgeinternational.org/support **(F)**
1.1
March 2019 Paper 12, question 3(a)
June 2019 Paper 11, question 1
June 2019 Paper 13, question 1(a)
Nov 2019 Paper 11, question 2
Nov 2019 Paper 12, question 4
Nov 2019 Paper 13, question 1(c)(i)
March 2020 Paper 12, question 5
June 2020 Paper 11, question 9(d)
June 2020 Paper 12, question 7
June 2020 Paper 13, question 1
2023 Specimen Paper 1, question 1
1.2 and 1.3
March 2019 Paper 12, question 1
June 2019 Paper 12, question 4(b)
Nov 2019 Paper 11, question 6(d)
Nov 2019 Paper 12, question 5(a) and 5(b)
Nov 2019 Paper 13, question 1(c)(ii)
March 2020 Paper 12, question 3(d)
June 2020 Paper 11, question 3(c)

Data transmission

Syllabus ref.	Learning objectives	Suggested teaching activities
2.1 Types and methods of data transmission	1 Understand that data is broken down into packets for transmitted; describe the structure of a packet; describe the process of packet switching.	Teacher introduces the use of packets for transmitting data and setting this in the context of the internet by showing https://www.youtube.com/watch?v=LGnJw9rtjas a short film. Teacher leads discussion transmission of data including: • breaking data into packets • packet structure of header (containing destination address, packet number and originator's address), payload and trailer • packet switching, the use of different routes and reordering. Notes on packets and packet switching: https://isaaccomputerscience.org/concepts/gcse_net_packet_switching?topic=gcse_networking
	2 Describe how data is transmitted from one device to another using different methods of data transmission. Explain the suitability of the method of data transmission for a given scenario.	Teacher leads discussion of what is meant by transmission of data; the need to check for errors; differences between serial and parallel data transmission. Notes on bandwidth: www.teach-ict.com/as_a2_ict_new/ocr/A2_G063/333_networks_coms/bandwidth/home_bandwidth.html Pages on serial, parallel, USB, etc.: http://computer.howstuffworks.com/computer-buses-channel.htm Example of simplex, duplex and half duplex: http://www.iec-usa.com/Browse05/DTHFDUP.html
	3 Understand the universal serial bus (USB) interface and how it is used to transmit data.	In pairs, learners identify current uses of serial and parallel data transmission e.g. Integrated Circuits (IC), Universal Serial Bus (USB); reasons for choosing serial or parallel data transmission. Learners answer previous exam/textbook questions on data transmission. **(F)**
2.2 Methods of error detection	1 Understand the need to check for errors after data transmission and how these errors can occur.	Teacher presentation to explain the methods of error detection and correction e.g. parity checks, check digits, checksums, Automatic Repeat reQuests (ARQ). Learners complete short case study scenario questions on the need to check for errors. **(F)**
	2 Describe the processes involved	Pages on error checking (checksum, CRC): http://computer.howstuffworks.com/encryption7.htm

Syllabus ref.	Learning objectives	Suggested teaching activities
	in error detection methods. 3 Describe how a check digit is used to detect errors in data entry including ISBN numbers and bar codes. 4 Describe how an automatic repeat request (ARQ) can be used to establish that data is received without error.	Teacher shows short video on check digits https://www.youtube.com/watch?v=tt9DYh6RFrY Give learners numbers to create the check digit and check that a given check digit is correct. Then get learners to set up a number with a correct check digit and another with an incorrect check digit and swap with another learner to find which number is correct. Notes on ARQ: http://en.wikipedia.org/wiki/Automatic_repeat_request Learners complete a multiple-choice quiz and answer previous exam/textbook questions on error detection.**(F)**
2.3 Encryption	1 Understand the need for and purpose of encryption when transmitting data. 2 Understand how data is encrypted using symmetric and asymmetric encryption.	Learners carry out coding and decoding activities in pairs use of symmetric encryption (plain text, cypher text, use of a key) – differentiation by task will make this more challenging where necessary. Information on encryption: https://www.101computing.net/symmetric-vs-asymmetric-encryption/ Learners attempt to decode some encrypted messages from the Central Intelligence Agency (CIA). **(I)** Class brainstorm code-breaking and relative ease of breaking each method. Notes on encryption: www.igcseict.info/theory/4/secure/index.html CIA code-breaking for kids: https://www.cia.gov/kids-page/games/break-the-code/code-1.html

Past and specimen papers

Past/specimen papers and mark schemes are available to download at www.cambridgeinternational.org/support **(F)**
March 2019 Paper 12, question 4(d) and question 6(a)
June 2019 Paper 12, question 9
June 2019 Paper 13, question 3(b) and question 5
Nov 2019 Paper 11, question 5
Nov 2019 Paper 12, question 5(c)(ii), question 9 and question 10(a)
Nov 2019 Paper 13, question 1(b) question 3 and question 5
March 2020 Paper 12, question 2
June 2020 Paper 11, question 3(a), 3(b) and question 4
June 2020 Paper 12, question 3
June 2020 Paper 13, question 9
2023 Specimen Paper 1, question 2(a), 2(b) and question 9

Hardware

Syllabus ref.	Learning objectives	Suggested teaching activities
3.1 Computer architecture	1 Understand the role of the central processing unit (CPU) in a computer. Understand what is meant by a microprocessor. 2 Understand the purpose of the components in a CPU, in a computer that has a Von Neumann architecture. Describe the role of each component in the process of the fetch–decode–execute cycle. 3 Understand what is meant by CPU core, cache and clock how they can affect the performance of a CPU. 4 Understand the purpose and use of a CPU instruction set. 5 Describe the purpose and characteristics of an embedded system and identify devices in which they are commonly used.	Teacher presents basic concepts of computer architecture, including registers, and the fetch-decode-execute cycle followed by demonstration via projector of the LMC. Teacher demonstrates the fetch-decode-execute cycle using a suitable video / animation a short simple introduction https://www.youtube.com/watch?v=04UGopESS6A a more in depth demonstration https://www.youtube.com/watch?v=jFDMZpkUWCw Notes on fetch-execute cycle: https://www.computerscience.gcse.guru/theory/fetch-execute-cycle https://teachcomputerscience.com/gcse/computer-architecture/functions-and-characteristics-of-cpu/ Teacher leads discussion on CPU performance: Useful notes: https://www.teach-ict.com/2016/GCSE_Computing/OCR_J276/1_1_systems_architecture/performance/miniweb/index.php Learners carry out simple low level tasks using LMC software – paired work is probably most effective. Little Man Computer web based: www.yorku.ca/sychen/research/LMC/ Teacher leads discussion of what is meant by an embedded system. Learner work in pairs to identify devices with embedded systems. Useful notes: https://learnlearn.uk/gcsecs/embedded-systems/ Learners answer flash cards, quiz and previous exam/textbook questions on computer architecture. **(F)** Flash cards: https://revisecomputerscience.com/courses/ocr-gcse-complete/lessons/function-of-the-cpu/topic/check-your-understanding-function-of-the-cpu/ Quiz: https://quizizz.com/admin/quiz/5ce2665f26c4eb001a8a86d2/ocr-cs-gcse-11-systems-architecture
3.2 Input and output devices	1 Understand what is meant by an input device and why it is required.	Start with a class discussion about what learners already know. What computer systems do they know about and where are they used? Write ideas on the whiteboard and expand with more examples.

Syllabus ref.	Learning objectives	Suggested teaching activities
	2 Understand what is meant by an output device and why it is required.	Ask learners to work in pairs to identify as many input and output devices for a computer system that they can write up on board and then add to that list. Develop this list by brainstorming different categories to include: • input devices (keyboard, optical mouse, barcode scanner, QR code scanner, touch screen (resistive, capacitive and infra-red), microphone, 2D scanner and 3D scanner) • output devices (printer (inkjet, laser and 3D), screen (LED and LCD), projector (DLP and LCD), speaker and actuator (note: the actuator could be discussed later in 6.1 automated systems) Learners are divided into small groups; each group investigates one specific input device and produces a report/poster/leaflet on: • typical applications • why it is used in these applications; its advantages and limitations. Reports/posters/leaflets to be prepared using software such as DTP, presentation software, intranet/internet pages etc. Each group gives a five-minute presentation on their input device. Learners work in different groups to investigate one specific output device and produce a report / poster / leaflet on: • typical applications • why it is used in these applications; its advantages and limitations. Reports / posters / leaflets to be prepared using software such as DTP, presentation software, intranet / internet pages etc. Each group gives a five-minute presentation on their output device. Each group prints enough copies of their input and output reports to provide a copy for each learner. If learner work is stored on an intranet, copy files into each learner account. Useful for research: http://computer.howstuffworks.com/
	3 Understand what is meant by a sensor and the purpose of sensors. Identify the type of data captured by each sensor, and when each sensor would be used, including selecting the	Teacher introduces the use of sensors for data capture. Sensors will be revisited in 6.1 Teacher introduction includes identification and uses for these sensors: • acoustic, accelerometer, flow, gas, humidity, infra-red, level, light, magnetic field, moisture, pH, pressure proximity and temperature Introduction to sensors and their uses: https://www.computerscience.gcse.guru/theory/sensors

Syllabus ref.	Learning objectives	Suggested teaching activities
	most suitable sensor for a given context.	Learners answer flash cards, quiz and previous exam/textbook questions on input devices, output devices and sensors. **(F)** https://www.computerscience.gcse.guru/quiz/input-devices https://www.computerscience.gcse.guru/quiz/output-devices
3.3 Data storage	1 Understand what is meant by primary storage. 2 Understand what is meant by secondary storage. 3 Describe the operation of magnetic, optical and solid-state storage and give examples of each. 4 Describe what is meant by virtual memory, how it is created, used and why it is necessary. 5 Understand what is meant by cloud storage. 6 Explain the advantages and disadvantages of storing data on the cloud in comparison to storing it locally.	Class brainstorms the difference between primary and secondary storage. Teacher leads discussion of data storage using physical examples and/or videos/animations showing operation including: • primary (RAM and ROM) • secondary • operation and examples of use of magnetic, optical and flash memory • virtual memory • cloud storage including the advantages and disadvantages of storing data on the cloud in comparison to storing it locally. Some sample videos: https://www.youtube.com/watch?v=FhmjWR9Y8Zk https://student.craigndave.org/videos/ocr-gcse-j277-slr-1-2-virtual-memory https://www.youtube.com/watch?v=iLnrilOZJ44 Learners complete gapped handouts on the operation and examples of use of magnetic, optical and flash memory. **(I)** Learners work in groups to describe the ways in which different types of memory are used in a typical day in their life (e.g. DVD/ Blu-ray to watch a film, USB flash memory to carry data to and from school, CD to play music, use of mobile, MP3 player etc.). Use an appropriate way to keep notes on their work. http://computer.howstuffworks.com/computer-ram-memory-channel.htm
3.4 Network hardware	1 Understand that a computer needs a network interface card (NIC) to access a network. 2 Understand what is meant by the	What is a network? Class brainstorms the concepts of networking and how devices on a network connect and communicate. Teacher introduces the purpose of MAC and IP addresses including:

Syllabus ref.	Learning objectives	Suggested teaching activities
	purpose and structure of a media access control (MAC) address, including its structure. 3 Understand what is meant by the purpose of an internet protocol (IP) address. Understand that there are different types IP address. 4 Describe the role of a router in a network.	• creation and structure of MAC addresses written as hexadecimal (relates back to section 1.1 Number Systems) • allocation of static and dynamic IP addresses, characteristics and differences between IPv4 and IPv6 addresses, use of a router to assign IP addresses Learner work in pairs to further research the role of a router. Useful video: https://www.youtube.com/watch?v=NOS0grrM9N4 Learners answer flash cards, quiz and previous exam/textbook questions on network hardware. **(F)**

Past and specimen papers

Past/specimen papers and mark schemes are available to download at www.cambridgeinternational.org/support **(F)**
3.1
June 2019 Paper 13, question 2(b)
Nov 2019 Paper 11, question 8
Nov 2019 Paper 12, question 2
Nov 2019 Paper 13, question 7(a)
March 2020 Paper 12, question 1
June 2020 Paper 11, question 5
June 2020 Paper 12, question 1
June 2020 Paper 13, question 5(b)(ii)
2023 Specimen Paper 1, question 4
3.2 (Any questions referring to the operation of an input/output device are now beyond the scope of this syllabus) and 3.3
March 2019 Paper 12, question 3(b), and question 6
June 2019 Paper 11, question 3 and question 6(b)
June 2019 Paper 12, question 4, question 7(b) and question 10
June 2019 Paper 13, question 6
Nov 2019 Paper 11, question 1 and question 4
Nov 2019 Paper 12, question 6, question 7(a) and question 11
Nov 2019 Paper 13, question 1(a) and 1(d)
March 2020 Paper 12, question 3(a), 3(b) and 3(c)
June 2020 Paper 11, question 1(a), 1(b), 1(d) question 6 and question 8

Syllabus ref.	Learning objectives	Suggested teaching activities
		June 2020 Paper 12, question 6, question 8(b) and question 9 June 2020 Paper 13, question 2, question 5(b)(i) and question 8
3.4		June 2019 Paper 13, question 2(a) June 2020 Paper 11, question 1(c) 2023 Specimen Paper 1, question 1(b), question 2 and question 3(a)

Software

Syllabus ref.	Learning objectives	Suggested teaching activities

4.1 Types of software and interrupts	1 Describe the difference between system software and application software and provide examples of each.
	2 Describe the role and basic functions of an operating system.
	3 Understand how hardware, firmware and an operating system are required to run applications software.
	4 Describe the role and operation of interrupts.
		Teacher presentation to include:
		• the idea of system software as different from applications software
		• role and basic functions of an operating system – it is helpful to demonstrate Windows
		• providing an interface, a platform for running applications and system security
		• managing files, multitasking, memory and user accounts
		• handling interrupts
		
		Several pages describing operating systems and their functions:
		www.howstuffworks.com/operating-system1.htm
		https://igcse.at.ua/IGCSE/ICT/notes-yusuf/Section-1-Types-and-Components-of-Computer-Systems.pdf
		http://www.kierenreynolds.com/quiz/uploads/GCSE/Unit5Softwareandsecurity/Lesson2Operatingsystems/SoftwareL2Operatingsystems.pdf
		
		Learners (paired/grouped) to research:
		• firmware
		• bootloader
		• interrupt generation and handling
		• software interrupts including:
		  ○ division by zero
		  ○ two processes trying to access the same memory location
		• hardware interrupts including:
		  ○ pressing a key on the keyboard
		  ○ moving the mouse
		
		Learners use their findings to create a short role play activity that demonstrates how each of these works
		Example activity: https://sites.google.com/site/childrenandtechnology/presentation-3-simulate-computer
		Learners need to make their own notes on each of these after they have been acted out. **(I)**
		
		Learners answer flash cards, quiz and previous exam/textbook questions on types of software and

Syllabus ref.	Learning objectives	Suggested teaching activities
		interrupts. **(F)** https://quizlet.com/511077964/igcse-computer-science-memory-and-data-storage-flash-cards/ www.teach-ict.com/gcse_computing/ocr/213_software/user_interface/home_user_interface.htm
4.2 Types of programming language, translators and integrated development environments (IDEs)	1 Explain what is meant by a high-level language, and low-level language, including the advantages and disadvantages of each. 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 a high-level language is translated by each and how errors are reported. 4 Explain the advantages and disadvantages of a compiler and an interpreter.	Teacher introduces learners to different types of programming languages by showing examples of programs written in machine code, assembly language and a high-level language. Teacher uses these to lead a discussion on advantages and disadvantages of each including: • ease of reading, writing and debugging code • machine independence • direct manipulation of hardware Learners complete a chart of advantages and disadvantages.**(I)** Teacher introduces learners to assembly language by considering: • historical origins of computer programming in machine-specific types of language (machine code and assembly language) • the characteristics of these languages • the need for an assembler translation program for assembly language Learners research the characteristics of high-level languages; the need for compiler and/or interpreter translation programs for these languages; why they are preferred for many applications. **(I)** Introduction to high-level language: www.teach-ict.com/gcse_computing/ocr/216_programming/ Highlight the differences between compilation and interpretation including: • **compiler** translates the whole program (source code) into an executable file that can be stored and re-used, an error report is provided for the whole code if errors are detected • **interpreter** translates and executes a program line-by-line, an interpreter stops execution when an error is found, no object code is stored for further use – a program has to be translated each time it is used. Discuss the advantages and disadvantages of compilation and interpretation highlighting when it would be appropriate to use a compiler or an interpreter (use an interpreter during program development as errors can be easily checked and modified and use a compiler to translate the final program to be distributed for use without the need of a translator).

Syllabus ref.	Learning objectives	Suggested teaching activities
		As learners have used translators they should be able to contribute to a discussion. Create a table with some statements about compilers and some statements about interpreters. Ask your learners to read the statements and tick which apply to a compiler and which apply to an interpreter. **Resources:** Link to theory notes on compilers and interpreters: www.teach-ict.com/as_as_computing/ocr/H447/F453/3_3_2/translators_compilers/miniweb/pg14.htm Short notes and exercises on program translators: http://en.wikibooks.org/wiki/A-level_Computing/AQA/Computer_Components,_The_Stored_Program_Concept_and_the_Internet/Fundamentals_of_Computer_Systems/Types_of_program_translator
	5 Explain the role of an IDE and the common functions it provides.	Teacher leads a discussion on the role of an IDE, as learners have used an IDE they should be able to contribute to a discussion. Teacher demonstrates or uses learners to demonstrate these functions: code editor, run-time environment, translators, error diagnostics, auto-completion, auto-correction and prettyprint. Learners answer previous exam/textbook questions on translators and IDEs. **(F)**

Past and specimen papers
Past/specimen papers and mark schemes are available to download at www.cambridgeinternational.org/support **(F)** March 2019 Paper 12, question 4(a) and 4(b) June 2019 Paper 11, question 7 Nov 2019 Paper 11, question 6(a) Nov 2019 Paper 12, question 7(a) Nov 2019 Paper 13, question 2 and question 7(b) March 2020 Paper 12, question 4 June 2020 Paper 11, question 8(a), question 9(a), 9(b) and 9(c) June 2020 Paper 12, question 2 and question 8(c) June 2020 Paper 13 2023 Specimen Paper 1, question 4(d) and question 5

The internet and its uses

Syllabus ref.	Learning objectives	Suggested teaching activities
5.1 The internet and the world wide web	1 Understand the difference between the internet and the world wide web. 2 Understand what is meant by a uniform resource locator (URL). 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. 5 Describe how web pages are located, retrieved and displayed of on a device when a user enters a URL. 6 Explain what is meant by cookies and how they are used, including session cookies and persistent cookies.	Class brainstorm/ teacher input to enable learners to: • Distinguish between the internet and the world wide web • describe the nature of the internet as a worldwide collection of computer networks. • explain the need to use a uniform resource locator (URL) and a web browser find and display web pages. • explain the importance and use of HTTP/HTTPS and cookies. Quiz - https://www.computerscience.gcse.guru/quiz/the-internet **(F)** Video – the internet: https://www.youtube.com/watch?v=5-IY6UGZF4s Learners (paired/grouped) to research and demonstrate functions of a web browser including: • storing bookmarks and favourites • recording user history • using multiple tabs • storing cookies • providing navigation tools and an address bar Teacher provides a gapped handout on the location, retrieval, and display of web pages, including the roles of the web browser, IP addresses, the DNS, web servers and HTML Teacher leads a discussion on the use of cookies including using them for saving personal and login details, tracking user preferences and holding items in an online shopping cart. Teacher explains when it is appropriate to use session cookies and persistent cookies. It could be useful to look at a cookie policy for example https://www.ocr.org.uk/about/our-policies/website-policies/cookie-policy/ Some useful notes: https://www.computerscience.gcse.guru/theory/cookies https://teachcomputerscience.com/cookies/

Syllabus ref.	Learning objectives	Suggested teaching activities
		Learners answer previous exam/textbook questions on the internet and the world wide web. **(F)**
5.2 Digital currency	1 Understand the concept and use of digital currency. 2 Understand the process of blockchain and how it is used to track digital currency.	Teacher introduces learners to digital currency and the use of block chain using an appropriate video for example: https://www.pwc.com/us/en/industries/financial-services/fintech/bitcoin-blockchain-cryptocurrency.html https://www.youtube.com/watch?v=8NgVGnX4KOw https://www.wyzowl.com/ico-cryptocurrency-videos/ https://www.youtube.com/watch?v=3xGLc-zz9cA Learners answer flash cards, quiz and textbook questions on digital currency. **(F)**
5.3 Cyber security	1 Describe the processes involved in, and the aim of carrying out a range of cyber security threats. 2 Explain how a range of solutions are used to help keep data safe from security threats.	Learners work in groups to research and create a leaflet/web pages on internet safety, addressing: • Malware (virus, worm, Trojan horse, spy-ware, adware and ransomware), hacking, brute force and denial of service attacks, data interception, phishing, pharming and social engineering and report the internet risks associated with these • the solutions available to help minimise the risks including: ○ how anti-malware (anti-virus and anti-spyware) and other protection software help keep data safe ○ authentication (username, password, biometrics and two-step verification) ○ access levels and privacy settings ○ checking URLs and the spelling and tone of a communication. Notes on biometrics: https://www.ictlounge.com/html/emerging_technologies.htm#biometrics Notes on network protection: www.teach-ict.com/gcse_new/protecting_systems/protecting_systems/miniweb/pg10.htm Information on computer security: http://computer.howstuffworks.com/computer-internet-security-channel.htm Notes on phishing and pharming: www.igcseict.info/theory/6/internet/index.html Teacher input to describe technical/practical issues around use of firewalls both software and hardware, including proxy servers; use of Secure Socket Layer (SSL).

Syllabus ref.	Learning objectives	Suggested teaching activities
		Learners answer flash cards, quiz and previous exam/textbook questions on cyber security. **(F)** Sample quiz: http://quizlet.com/10713330/phishing-and-pharming-flash-cards/

Past and specimen papers

Past/specimen papers and mark schemes are available to download at www.cambridgeinternational.org/support **(F)**
March 2019 Paper 12, question 7
June 2019 Paper 11, question 4 and question 6(a)
June 2019 Paper 12, question 5, question 6(a), 6(b), 6(d) and 6(e)
June 2019 Paper 13, question 1(d), 1(e) and question 3
Nov 2019 Paper 11, question 3 and question 6(b)
Nov 2019 Paper 12, question 7 and question 10
Nov 2019 Paper 13, question 8
March 2020 Paper 12, question 8
June 2020 Paper 11, question 1(e), question 7 and question 9(b)
June 2020 Paper 12, question 5 and question 10
June 2020 Paper 13, question 3, question 7 and question 10
2023 Specimen Paper 1, question 2(c) and question 10

Automated and emerging technologies

Syllabus ref.	Learning objectives	Suggested teaching activities
6.1 Automated systems	1 Describe how sensors, microprocessors and actuators can be used to create automated systems. 2 Describe the advantages and disadvantages of automated systems used for a given scenario.	Teacher introduction explaining the distinction between: • monitoring, in which system acquires data at intervals from sensors and, where necessary, analogue-to-digital converters (ADCs) and how software processes the input data to provide the user with information for monitoring physical or chemical quantities (such as temperature, flow rate or oxygen concentration) and warning signals if stored limits are exceeded. • control, in which input data may also be used as feedback from a system being controlled so that software can compare feedback with stored set-points or upper and lower limits to decisions about the outputs required to, where necessary, digital-to-analogue converters (DACs) and actuators, such as heaters or motorised valves. Learners work in groups; each group investigates one specific sensor from those listed in 2.1.3 and produces a report / poster / leaflet on: • how it works • typical scenarios where the sensor is used, chosen from industry, transport, agriculture, weather, gaming, lighting and science • why it is used in these scenarios; its advantages and disadvantages. Reports/posters/leaflets to be prepared using software such as DTP, presentation software, intranet/internet pages etc. Each group gives a five-minute presentation on their device. Each group prints enough copies of their reports to provide a copy for each learner. If learner work is stored on an intranet, copy files into each learner account. Illustrated notes on sensors: www.igcseict.info/theory/2/sensor/
6.2 Robotics	1 Understand what is meant by robotics. 2 Describe the characteristics of a robot.	Teacher introduction explaining robotics and the characteristics of a robot including : • mechanical structure • electrical components • programmable.

Syllabus ref.	Learning objectives	Suggested teaching activities
	3 Understand the roles that robots can perform and the advantages and disadvantages of their use.	Introduction to robotics: https://www.youtube.com/watch?v=wDsjcA6D3zA https://www.ictlounge.com/html/emerging_technologies.htm#robotics Learners work in groups; each group investigates one robot and produces a report / poster / leaflet on: • typical areas where the robot is used, chosen from industry, transport, agriculture, medicine, domestic and entertainment • the roles the robot performs, its advantages and disadvantages of its use. Reports/posters/leaflets to be prepared using software such as DTP, presentation software, intranet/internet pages etc. Each group gives a five-minute presentation on their device. Each group prints enough copies of their reports to provide a copy for each learner. If learner work is stored on an intranet, copy files into each learner account.
6.3 Artificial intelligence	1 Understand what is meant by artificial intelligence (AI). 2 Describe the main characteristics of AI. 3 Explain the basic operation and components of AI systems to simulate intelligent behaviour.	Teacher introduction explaining artificial intelligence (AI) and the main characteristics of AI including: • collection of data • ability to reason • ability to learn and adapt. Introduction to AI: https://www.ictlounge.com/html/emerging_technologies.htm#AI Teacher presentation to on the basic operation and components of AI systems limited to: • expert systems • machine learning Introduction to expert systems: https://www.ictlounge.com/html/expert_systems.htm Machine learning use the introduction here: https://teachinglondoncomputing.org/machine-learning/

Past and specimen papers

Past/specimen papers and mark schemes are available to download at www.cambridgeinternational.org/support **(F)**
March 2019 Paper 12, question 3(b)
June 2019 Paper 12, question 7(a)
Nov 2019 Paper 11, question 2
Nov 2019 Paper 12, question 4
Nov 2019 Paper 13, question 6
March 2020 Paper 12, question 7
June 2020 Paper 11, question 6
2023 Specimen Paper 1 Question 6

Algorithm design and problem-solving

Syllabus ref.	Learning objectives	Suggested teaching activities
7 Algorithm design and problem solving	1 Understand the program development life cycle: analysis, design, coding and testing.	Teacher introduction to the program development life cycle including: • analysis – identification, abstraction, decomposition of the problem and identification of requirements • design – shown by decomposition using structure diagrams, flowcharts and pseudocode • coding – writing program code and performing iterative testing • testing – testing program code using test data
	2 Understand that every computer system is made up of sub-systems, which are made up of further sub-systems. Understand how a problem can be decomposed into its component parts. Use different methods to design and construct a solution to a problem.	Class brainstorms a non-computer system to show that it is comprised of sub-systems. Learners analyse a relevant and appropriate system to identify sub-systems, and to sub-divide these. This could be the academic or personnel structure of a school/college, a department store, a large company, etc. Structure diagrams can be used to document this. **(I)** A tutorial on how to draw structure diagrams using SmartDraw, of which a free, trial version is available at: www.smartdraw.com/resources/tutorials/ Learners are introduced to the need for algorithms in developing solutions to problems. In pairs they can identify the sequence of operations required to carry out a simple multi-stage everyday process such as making a cup of tea/coffee, preparing a meal, etc. Teacher leads discussion showing how each problem can be decomposed into inputs, processes, outputs and storage. Teacher introduces the use of flowcharts and pseudocode to design and construct solutions. Teacher presents a simple flowchart using the symbols from the syllabus on page 31 to show: • flow of control/ data • explanation of terminator, input/output, process and decision boxes. Teacher presents a pseudocode algorithm for the same problem using the pseudocode guide from the syllabus on pages 33 to 46.

Syllabus ref.	Learning objectives	Suggested teaching activities
		Teacher introduces: • the use of variable names and mapping of values (e.g. x ← 3 means the value 3 is written as the new value stored in the memory location labelled x, x ← y means the value stored in the memory location labelled y is copied to the memory location labelled x) • use of mathematical operators including +, -, *, /,^, MOD and DIV • use of logical operators including =, <, >=, <=, >, and <> • use of Boolean operators including AND, OR and NOT
	3 Explain the purpose of a given algorithm.	Learners carry out analysis of prepared flowcharts to work out the purpose and the processes involved. The difficulty/complexity of the flowchart can be increased to make it more challenging where necessary. **(I)** Learners carry out analysis of prepared pseudocode to work out the purpose and the processes involved. **(I)** The difficulty/complexity of the pseudocode can be increased to make it more challenging where necessary. Learners answer previous exam/textbook questions on the program lifecycle and sub systems. **(F)**
	4 Understand standard methods of solution.	Teacher demonstrates standard methods of solution that can be used in solving many different problems including: • linear search and bubble sort • totalling and counting • finding maximum, minimum and average values Useful introductory videos: Linear search https://www.youtube.com/watch?v=PDE8pCQ9Tz4 Bubble sort https://www.youtube.com/watch?v=JP5KkzdUEYI Learners identify the standard methods of solution shown in teacher prepared flowcharts and pseudocode algorithms. Learners complete further gapped handouts showing the standard methods of solution. **(I)**
	5 Understand the need for validation checks to be made	Teacher explains need for validation checks to prevent input of incorrect data; teacher provides list of common applications (e.g. car registrations, test marks, learner names, temperatures, salaries) and

Syllabus ref.	Learning objectives	Suggested teaching activities
	on input data and the different types of validation check. Understand the need for verification checks to be made on input data and the different types of verification check.	learners identify possible validation rules. Teacher adds those that the class has not identified (need to cover range checks, length checks, type checks, presence checks, format checks and check digits). Teacher demonstrates introduction of validation checks with decision boxes in a sample flowchart; learners add validation to existing flowcharts. Calculation of check digits using ISBN (for example) by teacher demonstration and learner completion of a selection provided. Brainstorm the importance of verification when data is transferred between media (design flowchart for double entry of e.g. password). Teacher how verification would take place including visual checks and double entry. Teacher demonstrates introduction of verification checks with extra pseudocode in a sample algorithm; learners add verification to existing pseudocode algorithms. Notes, quizzes and activities for data validation: www.teach-ict.com/gcse_computing/ocr/databases/validating/home_validating.htm www.klbict.co.uk/gcse/theory/5_3/5_3_3_valid_verif.htm Learners answer previous exam/textbook questions on validation and verification checks. **(F)**
	6 Suggest and apply suitable test data.	Brainstorm to consider the limits for input data in any system; identify possible different types of input data (normal, boundary, extreme and abnormal/erroneous). Learners identify examples of each type for a range of given situations, as test data. **(I)** Learners answer previous exam/textbook questions on test data. **(F)**
	7 Complete a trace table to document a dry-run of an algorithms.	Teacher demonstrates design and completion of trace table for dry runs of a simple flowchart. Learners carry out exercise on flowcharts (opportunity for differentiation by complexity of flowchart). Teacher demonstrates design and completion of trace table for dry runs of a simple pseudocode algorithm. Learners carry out exercise on pseudocode algorithms (opportunity for differentiation by complexity of algorithm). **(I)(F)**

Syllabus ref.	Learning objectives	Suggested teaching activities
	8 Identify and suggest ways of correcting of errors in algorithms.	Learners use trace tables to analyse flowcharts for incorrect algorithms and identify the source of the errors.**(I)** Learners answer previous exam/textbook questions on trace tables.**(F)**
	9 Write and amend algorithms for given problems or scenarios using, pseudocode, program code and flowcharts.	Learners should perform practical exercises to demonstrate solution design such as: • finding the average of a set of input numbers • finding largest and smallest numbers in a set of input numbers calculating the frequency distribution of ranges of numbers in a set of input numbers (e.g. when a series of temperatures T are input, how many are in each of the ranges $-20 \leq T < 0$, $0 \leq T < 20$ and $20 \leq T < 40$?) **(I)** Using a simple algorithm, represented as a paragraph of English or as a flowchart, learners convert this to pseudocode. Simple examples could be numerical problems such addition, subtraction, multiplication, division; more challenging examples could be set in the real world, such as the use of an ATM. **(I)** These could be followed by case studies and questions. For example, an automatic supermarket stock control system for calculating stock levels and automatically re-ordering items. **(I)**

Past and specimen papers

Past/specimen papers and mark schemes are available to download at www.cambridgeinternational.org/support **(F)**
March 2019 Paper 22, question 2, question 3 and question 5
June 2019 Paper 21, question 4
June 2019 Paper 22, question 2, question 3 and question 5
June 2019 Paper 23, question 2, question 3 and question 4
Nov 2019 Paper 21, question 1(e), question 2 and question 4
Nov 2019 Paper 22, question 2, question 3 and question 4
Nov 2019 Paper 23, question 5 and question 6
March 2020 Paper 22 question 2, question 3 and question 4
June 2020 Paper 21, question 2, question 3, question 4 and question 5
June 2020 Paper 22, question 1(b), question 3 and question 4
June 2020 Paper 23, question 3, question 4, question 5 and question 6
2023 Specimen Paper 2, question 2, question 4, question 5 and question 7

Programming

Syllabus ref.	Learning objectives	Suggested teaching activities
8.1 Programming concepts	1 Declare and use variables and constants. 2 Understand and use basic data types. 3 Understand and use input and output. 4 Understand and use the concepts of sequence; selection; iteration, totalling; counting; string handling; and arithmetic, logical and Boolean operators.	Teacher introduces concepts of constants and variables; brainstorm to identify basic data types. Learners obtain definitions of data types (integer, real, char, string and Boolean) from web research. Theory notes on data types: www.teach-ict.com/gcse_computing/ocr/216_programming/handling_data/home_handling_data.htm Class brainstorm to revise use of: • variable names, assignment commands (←) • mathematical operators including +, -, *, /,^, MOD and DIV • logical operators including =, <, >=, <=, >, and <>, • Boolean operators including AND, OR and NOT Teacher revisits pseudocode for: • input and output (e.g. `READ` and `PRINT`) • selection using conditional statements `IF` … `THEN` … `ELSE` … `ENDIF`, `CASE OF` … `OTHERWISE` … `ENDCASE` • iteration with count-controlled loops `FOR` … `NEXT`, pre-condition loops `DO WHILE` … `ENDWHILE` and post-condition loops `REPEAT` … `UNTIL` • counting and totalling Teacher introduces the use of a high-level programming language (Python, VB or Java) with a suitable development IDE. There are a range of IDEs that are free to download and use: https://www.python.org/downloads/ https://bluej.org/ https://visualstudio.microsoft.com/downloads/ Learners practice using the IDE with simple programs e.g. 'Hello World' **(I)** Tutorials for various programming languages: www.codecademy.com/learn

Syllabus ref.	Learning objectives	Suggested teaching activities
		Learners convert previous flowcharts and algorithms for counting and totalling to simple programs. **(I)** Stress the need to initialise variables before using a loop and to output results after exiting a loop. Learners use loops to input specific number of items to write a program to calculate the average. Learners identify which type of loop is most appropriate for this and other different scenarios; explain rationale to class, who agree (or otherwise) on choice and rationale. Teacher introduces the concept of string handling (page 39 of the syllabus for pseudocode operations) including: • length • substring • upper and lower Learners write programs with simple string handling routines e.g. checking the length of a password, converting input to all upper or lower case, checking for a sequence of letters **(I)**
	5 Understand and use nested statements. 6 Understand what is meant by procedures, functions and parameters; Define and use procedures and functions, with or without parameters; Understand and use local and global variables.	Teacher introduces nested selection and iteration; learners design algorithms using nested selection and iteration. Level of nesting is limited to three. **(I)** Teacher demonstrates the use of procedures and functions including: • definition • use / calling • use of local and global variables • use of parameters (maximum two parameters) useful notes: https://studyrocket.co.uk/revision/gcse-computer-science-aqa/computational-thinking-and-problem-solving/structured-programming http://www.teach-ict.co.uk/2016/GCSE_Computing/OCR_J276/2_2_programming_techniques/functions_subs/miniweb/index.php Learners convert their simple string handling programs to functions without parameters (converting to upper and lower), with parameters checking the length and checking for a sequence of letters. **(I)**

Syllabus ref.	Learning objectives	Suggested teaching activities
	7 Understand and use library routines. 8 Understand how to create a maintainable program.	Learners research the use of library routines finding out: • why they are used • common uses including MOD, DIV, ROUND and RANDOM Learners devise programs to use several library routines including MOD, DIV, ROUND and RANDOM **(I)** Teacher shows an example of a program that is difficult to understand and goes through the process of making it easier to understand and maintain by using: • meaningful identifiers for variables, constants, arrays, procedures and functions • making use of the inbuilt commenting feature of the chosen programming language to include relevant and appropriate comments • procedures and/or functions where necessary Learners revisit and improve programs they have already written. **(I)** Learners write programming solutions for more detailed problems, the use of pre-release material for the old IGCSE paper two is a good source of problems. **(I)** Learners answer previous exam/textbook questions on programming concepts. **(F)**
8.2 Arrays	1 Declare and use one-dimensional (1D) and two-dimensional (2D) arrays. 2 Understand the use of arrays. 3 Write values into and read values from an array using iteration.	Teacher leads an introduction to arrays, explaining • how to declare the size of one- and two-dimensional arrays; for example: `A[1:n] and A[1:n, 1:m]` • the use of index variables in arrays • reading values into a one-dimensional array using a `FOR … TO … NEXT` loop to increment the index variable. • reading values into a two-dimensional array using a nested `FOR … TO … NEXT` loop to increment the index variables. Learners amend a previous task to read a set of data into an array and calculate the average. **(I)** Learners write programs to read values from a data source (data statement, keyboard) into a specified array and calculate the average, largest and smallest values. **(I)** Teacher revisits the linear search and bubble sort algorithms and shows how arrays would be used in

Syllabus ref.	Learning objectives	Suggested teaching activities
		these. Learners develop, with help if required, there own bubble sort and linear search programs. Learners give a partner some test data and test each other's programs recording their results in a trace table. Notes on arrays: www.docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html www.teach-ict.com/2016/GCSE_Computing/OCR_J276/2_2_programming_techniques/arrays_solve_problems/miniweb/index.php Learners answer previous exam/textbook questions on arrays.**(F)**

8.3 File handling	1 Understand the purpose of storing data in files.
	2 Open, close and use a file for reading and writing.
		Teacher introduces the concept of file handling (page 46 of the syllabus for pseudocode operations) including:
		• purpose of storing data in files
		• opening and closing a file
		• reading and writing to a file
		  ○ single items of data
		  ○ a line of text
		
		Notes on file handling:
		https://www.teach-ict.com/2016/GCSE_Computing/OCR_J276/2_2_programming_techniques/file_handling_operations/miniweb/pg2.php
		
		Leaners write pseudocode algorithms to write a string to a file and read the string back from the file.
		Leaners convert their algorithms to a program, then run and test the program. **(I)**
		
		Learners answer textbook questions on arrays. **(F)**

Past and specimen papers
Past/specimen papers and mark schemes are available to download at www.cambridgeinternational.org/support **(F)** March 2019 Paper 22, question 1 and question 4 June 2019 Paper 21, question 1, question 2 and question 3 June 2019 Paper 22, question 1 and question 4 June 2019 Paper 23, question 1 and question 5 Nov 2019 Paper 21, question 1 Nov 2019 Paper 22, question 1, question 5 and question 6 Nov 2019 Paper 23, question 1 and question 4 March 2020 Paper 22, question 1 June 2020 Paper 21, question 1 and question 4 June 2020 Paper 22, question 1, question 2 and question 5 June 2020 Paper 23, question 1, question 2, question 3 and question 4 2023 Specimen Paper 2, question 1, question 3, question 8, question 9, question 10, question 12 and question 13

Databases

Syllabus ref.	Learning objectives	Suggested teaching activities
9 Databases	1 Define a single-table database from given data storage requirements. 2 Suggest suitable basic data types. 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.	Teacher leads a review of a range of databases (manual, printed, electronic). Refer back to data types (Topic 7); identify structure of each record, data type of each field, field size, and record size. Issues of coding data should also be discussed. Review of data types: www.teach-ict.com/gcse_new/data_info_knowledge/data_types/miniweb/index.htm Assess learners' understanding with a quiz or similar activity. A pre-prepared set of questions can be used, or groups of learners could produce their own questions to test others. **(F)** Class brainstorm to identify appropriate contexts for learners' own database design, relevant to their experience and interests e.g. school register, clubs, library, video hire, stock in small shop. Learners select a context, identify fields, calculate field lengths, data types in each field, consider coding and validation for each field. Learners create a sample database in a sensible context; each learner needs to add about 20 records (enough to search later), so a data identification/ collection exercise will be needed. This could be done as homework. **(I)** Use software to create this database. Learners will need instruction on using the software if they have not used it before. Discussion about coding data: www.teach-ict.com/gcse_new/data_info_knowledge/coding/home_coding.htm Quiz-type activities: http://quizlet.com/2647975/gcse-ict-database-keywords-flash-cards/ **(F)** Microsoft Access, OpenOffice Base: Videos to support first-time use of the software can be found at: www.youtube.com/watch?v=TVaH-cgNz98 Teacher introduces the use of structured query language (SQL) scripts and demonstrates how to use a simple script to query a single table database. Practice using SQL, needs limiting to the SQL in the syllabus: https://www.w3schools.com/sql/ More quiz-type activities: https://quizlet.com/gb/342774585/unit-10-databases-sql-gcse-ocr-9-1-flash-cards/ **(F)**

Past and specimen papers

Past/specimen papers and mark schemes are available to download at www.cambridgeinternational.org/support **(F)**
June 2019 Paper 21 Question 5(a) and 5(b)
June 2019 Paper 22 Question 5(a)
June 2019 Paper 23 Question 6(a)
Nov 2019 Paper 21 Question 5(a) and 5(b)
Nov 2019 Paper 22 Question 7(a)
Nov 2019 Paper 23 Question 7(a) and 7(b)
March 2020 Paper 22 Question 5(a)
June 2020 Paper 21 Question 6(a)
June 2020 Paper 22 Question 6(a)
June 2020 Paper 23 Question7(a)
2023 Specimen Paper 2 Question 11
Note: learners do not need to use QBE grids, so these parts of the past questions have been omitted. Teachers can rewrite the QBE parts of the questions to use SQL to provide extra example questions.

Boolean Logic

Syllabus ref.	Learning objectives	Suggested teaching activities
10 Boolean logic	4 Identify and use standard symbols for logic gates. 5 Define and understand the functions of the logic gates. 6 Use of logic gates to create given logic circuits. Complete a truth table and write a logic expression.	Teacher introduces the concepts of OR and AND by careful use of English in logical statements; linking TRUE and FALSE to binary states (1 and 0). Teacher demonstrates OR gate and AND gate by use of electrical model, projector presentation of simulation, or similar. Teacher develops the concept of truth table for OR gate; learners develop truth table for AND gate. Introduction, notes and activities at: https://cambridgegcsecomputing.org/boolean-logic#top Teacher demonstrates OR gate and AND gate by use of electrical model, projector presentation of simulation, or similar. Teacher develops the concept of truth table for OR gate; learners develop truth table for AND gate. A useful presentation on binary logic is: www.youtube.com/watch?v=_76g8EM4DVU Teacher introduces the standard symbols for both gates, see section 4 of the syllabus. Learners use appropriate hardware or simulation software to develop understanding of the functions of the NOT, NAND, NOR and XOR (EOR) logic gates. **(I)** All gates are limited to two inputs except NOT which is a single input gate. Simple logic simulator using standard symbols: http://logic.ly/demo/ Downloadable logic gate simulator: www.softpedia.com/get/Others/Home-Education/Logic-Gate-Simulator.shtml and www.logiccircuit.org/ Learners can observe the output produced from all possible combinations of inputs to construct each gate's truth table. **(I)** **Extension activity:** Learners work out the simple logic circuits required to create NAND and NOR gates using AND, OR and NOT gates and test them. **(I)**

Past and specimen papers

Past/specimen papers and mark schemes are available to download at www.cambridgeinternational.org/support **(F)**
March 2019 Paper 12, question 5
June 2019 Paper 11, question 2
June 2019 Paper 12, question 8

Syllabus ref.	Learning objectives	Suggested teaching activities
June 2019 Paper 13, question 4 Nov 2019 Paper 11, question 2 Nov 2019 Paper 12, question 4 Nov 2019 Paper 13, question 6 March 2020 Paper 12, question 6 June 2020 Paper 11, question 2 June 2020 Paper 12, question 4 June 2020 Paper 13, question 6 2023 Specimen Paper 2, question 6 Note: from 2023 Boolean logic questions will be set in paper 2.		

Cambridge Assessment International Education
The Triangle Building, Shaftesbury Road, Cambridge, CB2 8EA, United Kingdom
t: +44 1223 553554
e: info@cambridgeinternational.org www.cambridgeinternational.org