

IT & Computing

CIE Computer Science 0984

iGCSE ~ Revision Checklist ~ 2019 exams

This is a list of the main topic heading from the CIE 0984 Syllabus for examination in 2019 onwards.

The Syllabus is available on PupilData, and should also be on your own personal website.

You can also access the syllabus on:

<http://www.cambridgeinternational.org/programmes-and-qualifications/cambridge-igcse-9-1-ukonly-computer-science-0984/>

NB – The syllabus is dated for the year you take the exam.

Text Books & Resources

This is a rapidly changing and evolving course, and does make use out of a wide range of resources.

Use	Title	Publisher
Main text book	Computer Science – Coursebook	Cambridge University Press
Revision Guide	Computer Science – Revision Guide	Cambridge University Press
Programming Guide	Computer Science – Programming Book - Python	Cambridge University Press
Main website	https://www.cambridgegcsecomputing.org/	Cambridge University Press
Secondary website	www.igcseict.info	Igcseict.info
Secondary website	http://www.sqa.org.uk/e-learning/SiteHomeCD/page_27.htm	Scottish Qualifications Authority
Secondary website	www.teach-ict.com	Teach-ICT.com
Python programming	https://www.learnpython.org/	LearnPython.org

Assessments

Students will take 2 exams in the summer of the 5th Form.

Component 1 ~ Paper 1 Theory

Written paper, 1 hour 45 minutes, 75 marks

This is a compulsory question paper, consisting of short-answer and structured questions set on Section 1 of the Subject content. All questions are compulsory. Candidates answer on the question paper.

Component 2 – Paper 2 Problem-Solving and Programming

Written paper, 1 hour 45 minutes, 50 marks

This is a compulsory question paper, consisting of short-answer and structured questions set on Section 2 of the Subject content. All questions are compulsory. Candidates answer on the question paper.

20 of the marks in this paper are from questions set on tasks provided in the Paper 2 Problem Solving and Programming pre-release material. Candidates must program in a high-level programming language chosen by the Centre. The programming language must be procedural.

Paper 2 – Problem-solving and Programming pre-release material

The Paper 2 Problem-solving and Programming pre-release material is available to Centres shortly after the estimated entries deadline for the June and November examinations. It is also reproduced in the question paper. Candidates must not bring any prepared material into the examination.

Syllabus Topics





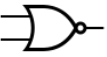
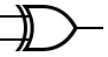
Use your textbook and the websites listed above to find the detailed information about each topic – then check it off when you have fully covered this topic.

NB – You should revise each topic at least 3 times to make sure you understand it – if in doubt please ask!

Section 1 ~ Theory of Computer Science

No	Topic	1	2	3
1.1	Data representation			
1.1.1	Binary systems You must be able to:			
	<ul style="list-style-type: none"> Recognise the use of binary numbers in computer systems 			
	<ul style="list-style-type: none"> Convert positive denary integers into binary and positive binary integers into denary (a maximum of 16 bits will be used) 			
	<ul style="list-style-type: none"> Show understanding of the concept of a byte and how the byte is used to measure memory size 			
	<ul style="list-style-type: none"> Use binary in computer registers for a given application (such as in robotics, digital instruments and counting systems) 			
1.1.2	Hexadecimal You must be able to:			
	<ul style="list-style-type: none"> Represent positive numbers in hexadecimal notation 			
	<ul style="list-style-type: none"> Show understanding of the reasons for choosing hexadecimal notation to represent numbers 			
	<ul style="list-style-type: none"> Convert positive hexadecimal integers to and from denary (a maximum of four hexadecimal digits will be required) 			
	<ul style="list-style-type: none"> Convert positive hexadecimal integers to and from binary (a maximum of 16 bit binary numbers will be required) 			
	<ul style="list-style-type: none"> Represent numbers stored in registers and main memory as hexadecimal 			
	<ul style="list-style-type: none"> Identify current uses of hexadecimal numbers in computing, such as defining colours in Hypertext Markup Language (HTML), Media Access Control (MAC) addresses, assembly languages and machine code, debugging 			
1.1.3	Data storage You must be able to:			
	<ul style="list-style-type: none"> Show understanding that sound (music), pictures, video, text and numbers are stored in different formats 			
	<ul style="list-style-type: none"> Identify and describe methods of error detection and correction, such as parity checks, check digits, checksums and Automatic Repeat reQuests (ARQ) 			
	<ul style="list-style-type: none"> Show understanding of the concept of Musical Instrument Digital Interface (MIDI) files, JPEG files, MP3 and MP4 files 			
	<ul style="list-style-type: none"> Show understanding of the principles of data compression (lossless and lossy) applied to music/video, photos and text files 			

No	Topic	1	2	3
1.2	Communication and Internet technologies			
1.2.1	Data transmission You must be able to:			
	<ul style="list-style-type: none"> Show understanding of what is meant by transmission of data 			
	<ul style="list-style-type: none"> Distinguish between serial and parallel data transmission 			
	<ul style="list-style-type: none"> Distinguish between simplex, duplex and half-duplex data transmission 			
	<ul style="list-style-type: none"> Show understanding of the reasons for choosing serial or parallel data transmission 			
	<ul style="list-style-type: none"> Show understanding of the need to check for errors 			
	<ul style="list-style-type: none"> Explain how parity bits are used for error detection 			
	<ul style="list-style-type: none"> Show understanding of the use of serial and parallel data transmission, in Universal Serial Bus (USB) and Integrated Circuit (IC) 			
1.2.2	Security aspects (This section links with section 1.4 of the syllabus.) You must be able to:			
	<ul style="list-style-type: none"> Show understanding of the security aspects of using the Internet and understand what methods are available to help minimise the risks 			
	<ul style="list-style-type: none"> Show understanding of the Internet risks associated with malware, including viruses, spyware and hacking 			
	<ul style="list-style-type: none"> Explain how anti-virus and other protection software helps to protect the user from security risks 			
1.2.3	Internet principles of operation You must be able to:			
	<ul style="list-style-type: none"> Show understanding of the role of the browser 			
	<ul style="list-style-type: none"> Show understanding of the role of an Internet Service Provider (ISP) 			
	<ul style="list-style-type: none"> Show understanding of what is meant by hypertext transfer protocol (http and https) and HTML 			
	<ul style="list-style-type: none"> Distinguish between HTML structure and presentation 			
	<ul style="list-style-type: none"> Show understanding of the concepts of MAC 			
1.3	Hardware and software			
1.3.1	Logic gates You must be able to:			
	<ul style="list-style-type: none"> Use logic gates to create electronic circuits 			

No	Topic	1	2	3
	<ul style="list-style-type: none"> Understand and define the functions of NOT, AND, OR, NAND, NOR and XOR (EOR) gates, including the binary output produced from all the possible binary inputs (all gates, except the NOT gate, will have 2 inputs only) 			
	<ul style="list-style-type: none"> Draw truth tables and recognise a logic gate from its truth table 			
	<ul style="list-style-type: none"> Recognise and use the following standard symbols used to represent logic gates: <div style="display: flex; justify-content: space-around; align-items: center; text-align: center;"> <div style="margin: 5px;">  NOT </div> <div style="margin: 5px;">  AND </div> <div style="margin: 5px;">  OR </div> <div style="margin: 5px;">  NAND </div> <div style="margin: 5px;">  NOR </div> <div style="margin: 5px;">  XOR </div> </div>			
	<ul style="list-style-type: none"> Produce truth tables for given logic circuits 			
	<ul style="list-style-type: none"> Produce a logic circuit to solve a given problem or to implement a given written logic statement 			
1.3.2	Computer architecture and the fetch-execute cycle You must be able to:			
	<ul style="list-style-type: none"> Show understanding of the basic Von Neumann model for a computer system and the stored program concept (program instructions and data are stored in main memory and instructions are fetched and executed one after another) 			
	<ul style="list-style-type: none"> Describe the stages of the fetch-execute cycle, including the use of registers and buses 			
1.3.3	Input devices You must be able to:			
	<ul style="list-style-type: none"> Describe the principles of operation (how each device works) of these input devices: <ul style="list-style-type: none"> ○ 2D and 3D scanners ○ Barcode readers ○ Quick Response (QR) code readers ○ Digital cameras ○ Keyboards ○ Mice ○ Touch screens ○ Interactive whiteboards ○ Microphones 			
	<ul style="list-style-type: none"> Describe how these principles are applied to real-life scenarios, for example: <ul style="list-style-type: none"> ○ Scanning of passports at airports ○ Barcode readers at supermarket checkouts ○ Touch screens on mobile devices 			

No	Topic	1	2	3
	<ul style="list-style-type: none"> • Describe how a range of sensors can be used to input data into a computer system, including: <ul style="list-style-type: none"> ○ Light ○ Temperature ○ Magnetic field ○ Gas ○ Pressure ○ Moisture ○ Humidity ○ pH ○ Motion 			
	<ul style="list-style-type: none"> • Describe how these sensors are used in real-life scenarios, for example: <ul style="list-style-type: none"> ○ Street lights ○ Security devices ○ Pollution control ○ Games ○ Household and industrial applications 			
1.3.4	Output devices You must be able to:			
	<ul style="list-style-type: none"> • Describe the principles of operation of the following output devices: <ul style="list-style-type: none"> ○ Inkjet, laser and 3D printers ○ 2D and 3D cutters ○ Speakers and headphones ○ Actuators ○ Flat-panel display screens, such as: <ul style="list-style-type: none"> • Liquid Crystal Display (LCD) • Light-Emitting Diodes (LED) display ○ LCD projectors ○ Digital Light Projectors (DLP) 			
	<ul style="list-style-type: none"> • Describe how these principles are applied to real-life scenarios, for example: <ul style="list-style-type: none"> ○ Printing single items on demand or in large volumes ○ Use of small screens on mobile devices 			
1.3.5	Memory, storage devices and media You must be able to:			

No	Topic	1	2	3
	<ul style="list-style-type: none"> • Show understanding of the difference between: primary, secondary and off-line storage and provide examples of each, such as: <ul style="list-style-type: none"> ○ Primary <ul style="list-style-type: none"> ▪ Read Only Memory (ROM) ▪ Random Access Memory (RAM) ○ Secondary <ul style="list-style-type: none"> ▪ Hard disk drive (HDD) ▪ Solid State Drive (SSD) ○ Off-line <ul style="list-style-type: none"> ▪ Digital Versatile Disc (DVD) ▪ Compact Disc (CD) ▪ Blu-ray disc ▪ USB flash memory ▪ Removable HDD 			
	<ul style="list-style-type: none"> • Describe the principles of operation of a range of types of storage device and media including: <ul style="list-style-type: none"> ○ Magnetic ○ Optical ○ Solid state 			
	<ul style="list-style-type: none"> • Describe how these principles are applied to currently available storage solutions, such as: <ul style="list-style-type: none"> ○ Digital Versatile Disc (DVD) ○ Compact Disc (CD) ○ Blu-ray disc ○ USB flash memory 			
	<ul style="list-style-type: none"> • Calculate the storage requirement of a file 			
1.3.6	Operating systems You must be able to:			
	<ul style="list-style-type: none"> • Describe the purpose of an operating system (Candidates will be required to understand the purpose and function of an operating system and why it is needed. They will not be required to understand how operating systems work.) 			
	<ul style="list-style-type: none"> • Show understanding of the need for interrupts 			
1.3.7	High- and low-level languages and their translators You must be able to:			
	<ul style="list-style-type: none"> • Show understanding of the need for both high-level and low-level languages 			
	<ul style="list-style-type: none"> • Show understanding of the need for compilers when translating programs written in a high-level language 			
	<ul style="list-style-type: none"> • Show understanding of the use of interpreters with high-level language programs 			
	<ul style="list-style-type: none"> • Show understanding of the need for assemblers when translating programs written in assembly language 			

No	Topic	1	2	3
1.4	Security			
1.4.1	Keeping data safe You must be able to:			
	<ul style="list-style-type: none"> Show understanding of the need to keep data safe from accidental damage, including corruption and human errors 			
	<ul style="list-style-type: none"> Show understanding of the need to keep data safe from malicious actions, including unauthorised viewing, deleting, copying and corruption 			
1.4.2	Security of data transfer You must be able to:			
	<ul style="list-style-type: none"> Show understanding of how data are kept safe when stored and transmitted, including: <ul style="list-style-type: none"> Use of passwords, both entered at a keyboard and biometric Use of firewalls, both software and hardware, including proxy servers Use of security protocols such as Secure Socket Layer (SSL) and Transport Layer Security (TLS) Use of symmetric encryption (plain text, cypher text and use of a key) showing understanding that increasing the length of a key increases the strength of the encryption 			
	<ul style="list-style-type: none"> Show understanding of the need to keep online systems safe from attacks including <ul style="list-style-type: none"> Denial of service attacks Phishing Pharming 			
	<ul style="list-style-type: none"> Describe how the knowledge from 1.4.1, 1.4.2 and 1.4.3 can be applied to real-life scenarios including, for example, online banking, shopping 			
1.5	Ethics You must be able to:			
	<ul style="list-style-type: none"> Show understanding of computer ethics, including: <ul style="list-style-type: none"> Copyright issues Plagiarism 			
	<ul style="list-style-type: none"> Distinguish between: <ul style="list-style-type: none"> Free software Freeware Shareware 			
	<ul style="list-style-type: none"> Show understanding of the ethical issues raised by the spread of electronic communication and computer systems, including hacking, cracking and production of malware 			

Section 2 ~ Practical Problem-Solving and Programming

No	Topic	1	2	3
2.1	Algorithm design and problem-solving			
2.1.1	Problem-solving and design You must be able to:			
	<ul style="list-style-type: none"> Show understanding that every computer system is made up of sub-systems, which in turn are made up of further sub-systems 			
	<ul style="list-style-type: none"> Use top-down design, structure diagrams, flowcharts, pseudocode, library routines and subroutines 			
	<ul style="list-style-type: none"> Work out the purpose of a given algorithm 			
	<ul style="list-style-type: none"> Explain standard methods of solution 			
	<ul style="list-style-type: none"> Suggest and apply suitable test data 			
	<ul style="list-style-type: none"> Understand the need for validation and verification checks to be made on input data (validation could include range checks, length checks, type checks and check digits) 			
	<ul style="list-style-type: none"> Use trace tables to find the value of variables at each step in an algorithm 			
	<ul style="list-style-type: none"> Identify errors in given algorithms and suggest ways of removing these errors 			
	<ul style="list-style-type: none"> Produce an algorithm for a given problem (either in the form of pseudocode or flowchart) 			
	<ul style="list-style-type: none"> Comment on the effectiveness of a given solution 			
2.1.2	Pseudocode and flowcharts You must be able to:			
	<ul style="list-style-type: none"> Understand and use pseudocode for assignment, using ← 			
	<ul style="list-style-type: none"> Understand and use pseudocode, using the following conditional statements: <ul style="list-style-type: none"> IF ... THEN ... ELSE ... ENDIF CASE ... OF ... OTHERWISE ... ENDCASE 			
	<ul style="list-style-type: none"> Understand and use pseudocode, using the following loop structures: <ul style="list-style-type: none"> FOR ... TO ... NEXT REPEAT ... UNTIL WHILE ... DO ... ENDWHILE 			
	<ul style="list-style-type: none"> Understand and use pseudocode, using the following commands and statements: <ul style="list-style-type: none"> INPUT and OUTPUT (e.g. READ and PRINT) Totalling (e.g. Sum ← Sum + Number) Counting (e.g. Count ← Count + 1) 			
	<ul style="list-style-type: none"> Understand and use standard flowchart symbols to represent the above statements, commands and structures 			

No	Topic	1	2	3
2.2	Programming			
2.2.1	Programming concepts You must be able to:			
	<ul style="list-style-type: none"> • Declare and use variables and constants 			
	<ul style="list-style-type: none"> • Understand and use basic data types: <ul style="list-style-type: none"> ○ Integer ○ Real ○ Char ○ String ○ Boolean 			
	<ul style="list-style-type: none"> • Understand and use the concepts of: <ul style="list-style-type: none"> ○ Sequence ○ Selection ○ Repetition ○ Totalling ○ Counting 			
	<ul style="list-style-type: none"> • Use predefined procedures/functions 			
2.2.2	Data structures; arrays You must be able to:			
	<ul style="list-style-type: none"> • Declare and use one-dimensional arrays, for example: A[1:n] 			
	<ul style="list-style-type: none"> • Show understanding of the use of one-dimensional arrays, including the use of a variable as an index in an array 			
	<ul style="list-style-type: none"> • Read or write values in an array using a FOR ... TO ... NEXT loop 			
2.3	Databases You must be able to:			
	<ul style="list-style-type: none"> • Define a single-table database from given data storage requirements 			
	<ul style="list-style-type: none"> • Choose and specify suitable data types 			
	<ul style="list-style-type: none"> • Choose a suitable primary key for a database table 			
	<ul style="list-style-type: none"> • Perform a query-by-example from given search criteria 			

NB – You should revise each topic at least 3 times to make sure you understand it – if in doubt please ask!