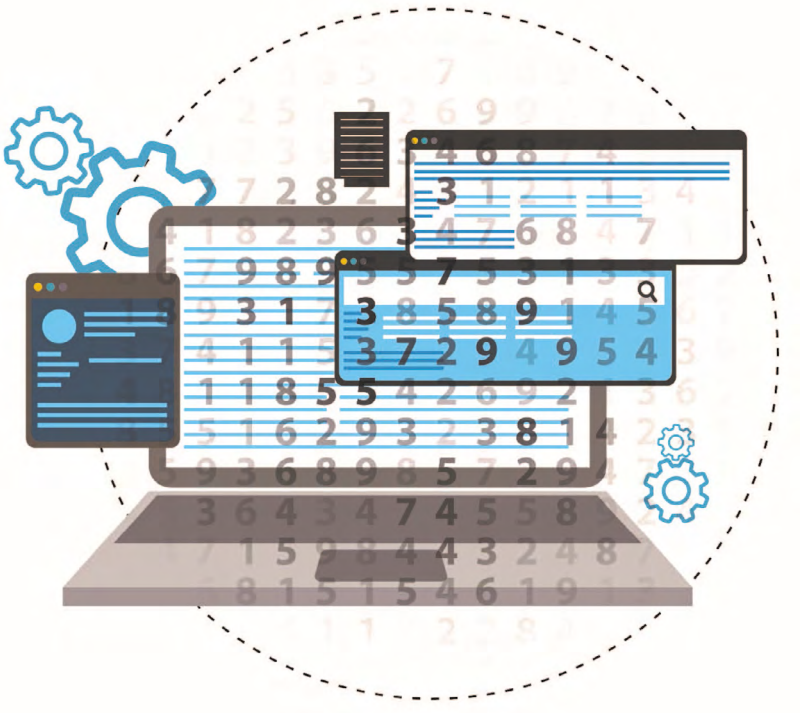
Cambridge Assessment International Education

Interactive

Learner Guide

Cambridge International AS & A Level Computer Science 9618

For examination from 2021



Version 1

Cambridge Pathway

In order to help us develop the highest quality resources, we are undertaking a continuous programme of review; not only to measure the success of our resources but also to highlight areas for improvement and to identify new development needs.

We invite you to complete our survey by visiting the website below. Your comments on the quality and relevance of our resources are very important to us.

www.surveymonkey.co.uk/r/GL6ZNJB

Contents

About this guide

This guide explains what you need to know about your Cambridge International AS & A Level Computer Science course and examinations.

This guide will help you to:

- understand what skills you should develop by taking this AS & A Level course
- understand how you will be assessed
- understand what we are looking for in the answers you write
- plan your revision programme
- revise, by providing revision tips and an interactive revision checklist (Section 7).

Following a Cambridge International AS & A level programme will help you to develop abilities that universities value highly, including a deep understanding of your subject; higher order thinking skills (analysis, critical thinking, problem solving); presenting ordered and coherent arguments; and independent learning and research.

Studying Cambridge International AS & A Level Computer Science will help you to develop a set of transferable skills, including computational thinking; problem solving; understanding communication methods; writing programs for computer based solutions; analysing and evaluating computer based systems and making reasoned judgements.

Our approach in Cambridge International AS & A Level Computer Science encourages learners to be:

confident, using a range of technology and programming paradigms

responsible, using technology ethically

reflective, as programmers, improving their own programming solution

innovative, creating efficient solutions to problems

engaged, in technology, how it is built and how software solutions are developed.

Section 1: Syllabus content - what you need to know

This section gives you an outline of the syllabus content for this course. There are four components that you will need to take.

Content section	Component	Topics included
Computer systems including software, data hardware and people	Paper 1 Theory Fundamentals	• Information representation • Communication • Hardware • Processor Fundamentals • System Software • Security, privacy and data integrity • Ethics and Ownership
Networks and Communication		• Networks including the internet
Databases		• Database concepts • Database Management system (DBMS), Data Definition Language (DDL) and Data Manipulation Language (DML)
Solving problems using computers	Paper 2 Fundamental Problem-solving and Programming Skills	• Algorithm Design and Problem-Solving • Data Types and Structures • Programming • Software Development
Computer systems including software, data hardware and people	Paper 3 Advanced Theory	• Data Representation • Hardware and Virtual Machines • Systems Software • Artificial Intelligence
Networks, Communication and Security		• Communication and internet technologies • Security
Computational thinking, problem-solving and programming	Paper 4 Practical	• Algorithms • Recursion • Programming Paradigms • File Processing and Exception Handling

Make sure you always check the latest syllabus, which is available from our public website.

Prior knowledge

Candidates beginning this course are not expected to have studied computer science previously. However, the content of the Cambridge IGCSE™ Computer Science 0478, Cambridge O Level Computer Science 2210 or Cambridge IGCSE (9–1) Computer Science 0984 will provide a useful introduction to the subject.

Key concepts

Key concepts are essential ideas that help you to develop a deep understanding of your subject and make links between different aspects of the course. The key concepts for Cambridge International AS & A Level Computer Science are:

- Computational thinking

  Computational thinking is a set of fundamental skills that help produce a solution to a problem. Skills such as abstraction, decomposition and algorithmic thinking are used to study a problem and design a solution that can be implemented. This may involve using a range of technologies and programming languages.

- Programming paradigms

  A programming paradigm is a way of thinking about or approaching problems. There are many different programming styles that can be used, which are suited to unique functions, tools and specific situations. An understanding of programming paradigms is essential to ensure they are used appropriately, when designing and building programs.

- Communication

  Communication is a core requirement of computer systems. It includes the ability to transfer data from one device or component to another and an understanding of the rules and methods that are used in this data transfer. Communication could range from the internal transfer of data within a computer system, to the transfer of a video across the internet.

- Computer architecture and hardware

  Computer architecture is the design of the internal operation of a computer system. It includes the rules that dictate how components and data are organised, how data are communicated between components, to allow hardware to function. There is a range of architectures, with different components and rules, that are appropriate for different scenarios.

  All computers comprise of a combination of hardware components, ranging from internal components, such as the Central Processing Unit (CPU) and main memory, to peripherals. To produce effective and efficient programs to run on hardware, it is important to understand how the components work independently and together to produce a system that can be used. Hardware needs software to be able to perform a task. Software allows hardware to become functional. This enables the user to communicate with the hardware to perform tasks.

- Data representation and structures

  Computers use binary and understanding how a binary number can be interpreted in many different ways is important. Programming requires an understanding of how data can be organised for efficient access and/or transfer.

These key concepts help you to gain:

- a greater depth as well as breadth of subject knowledge
- confidence, especially in applying your knowledge and skills in new situations
- the vocabulary to discuss the subject conceptually and show how different aspects link together
- a level of mastery of their subject to help them enter higher education.

Section 2: How you will be assessed

Cambridge International AS Level Computer Science makes up the first half of the Cambridge International A Level course in Computer Science and provides a foundation for the study of Computer Science at Cambridge International A Level.

About the examinations

There are three different combinations of papers you can take to obtain a Computer Science AS or A Level qualification:

- AS Level only: Paper 1 and Paper 2 are taken at the end of the first year.
- A Level: taken over two years with Paper 1 and Paper 2 taken at the end of the first year, completing the A Level exam Paper 3 and Paper 4 at the end of year two.
- A Level where all components are examined at the end of the two-year course.

These are summarised in the diagram. Find out from your teacher which route you will be taking.

	Route	Paper 1	Paper 2	Paper 3	Paper 4
1	**AS Level only** (You take all AS Level components in the same exam series)	✓	✓		
2	**A Level** (staged over two years) Year 1 AS Level	✓	✓		
	Year 2 Complete the A Level			✓	✓
3	**A Level** (You take all components in the same exam series)	✓	✓	✓	✓

About the papers

The table gives you further information about the examination papers:

Component	Time and marks	Questions	Percentage of qualification
Paper 1 Theory Fundamentals	1 hour 30 minutes 75 marks	Sections 1 to 8 of the syllabus content. You will answer all questions.	50% of the AS Level 25% of the A Level
Paper 2 Fundamental Problem-solving and Programming Skills	2 hours 75 marks	Sections 9 to 12 of the syllabus content. You will need to write answers in pseudocode. You will answer all questions.	50% of the AS Level 25% of the A Level
Paper 3 Advanced Theory	1 hour 30 minutes 75 marks	Sections 13 to 20 of the syllabus content. You will answer all questions.	25% of the A Level
Paper 4 Practical	2 hours 30 minutes 75 marks	Sections 19 to 20 of the syllabus content. You will need to write and test programs written in either Java, VB.NET or Python. You will have the use of a computer without internet or email facility. You will answer all questions.	25% of the A Level

Section 3: What skills will be assessed?

The examiners take account of the following skills areas (**assessment objectives**) in the examinations:

Assessment objectives (AO)	What does the AO mean?
AO1 Knowledge and understanding	Demonstrate knowledge and understanding of the principles and concepts of computer science including abstraction, logic, algorithms and data representation.
AO2 Application	Apply knowledge and understanding of the principles and concepts of computer science, including to analyse problems in computational terms.
AO3 Design, program and evaluation	Design, program and evaluate computer systems to solve problems, making reasoned judgements about these.

It is important that you know the different weightings (%) of the assessment objectives, as this affects how the examiner will assess your work.

Assessment objective	Weighting at AS Level %	Weighting at A Level %
AO1	30	30
AO2	40	30
AO3	30	40
Total	100	100

Assessment objectives as a percentage of each component

Assessment objective	Weighting in components %			
	Paper 1	**Paper 2**	**Paper 3**	**Paper 4**
AO1	60	–	60	–
AO2	40	40	40	–
AO3	–	60	–	100
Total	100	100	100	100

Section 4: Command words

The table below includes command words used in the assessment for this syllabus. The use of the command word will relate to the subject context.

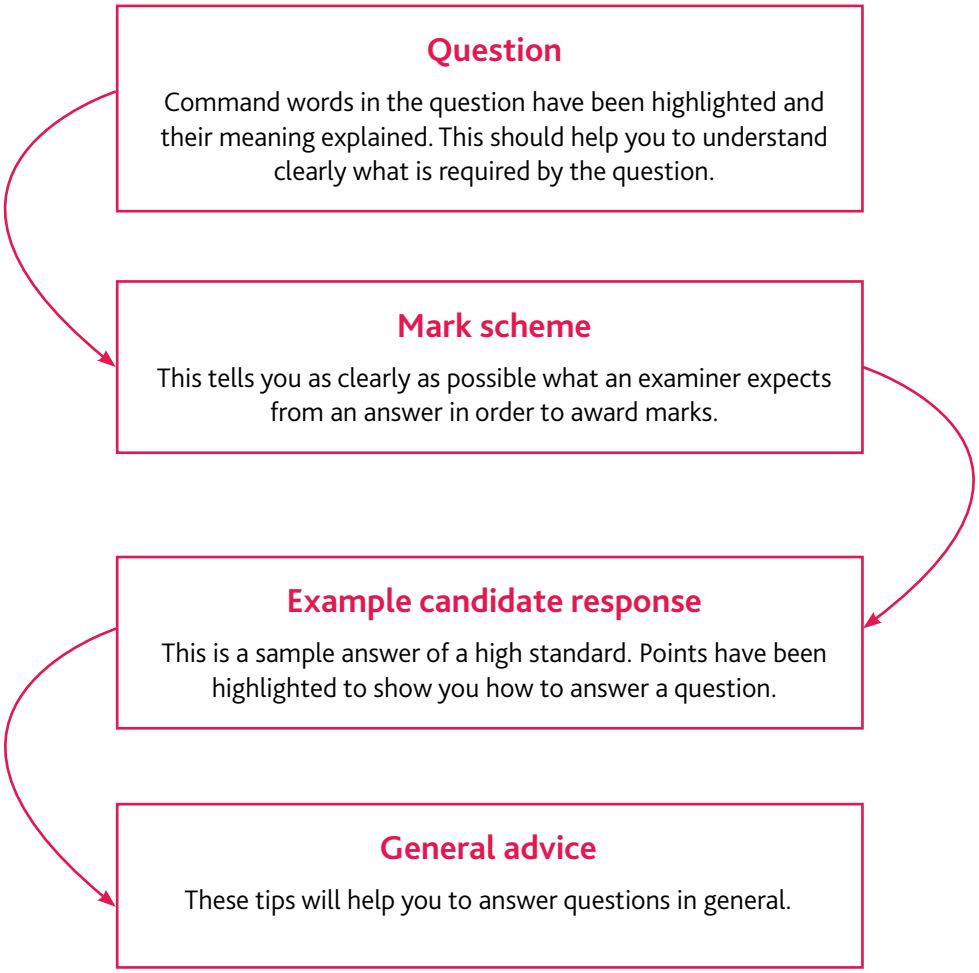
Command word	What it means
Analyse	examine in detail to show meaning, identify elements and the relationship between them
Assess	make an informed judgement
Calculate	work out from given facts, figures or information
Comment	give an informed opinion
Compare	identify/comment on similarities and/or differences
Complete	add information to an incomplete diagram or table
Consider	review and respond to given information
Contrast	identify/comment on differences
Define	give precise meaning
Demonstrate	show how or give an example
Describe	state the points of a topic / give characteristics and main features
Develop	take forward to a more advanced stage or build upon given information
Discuss	write about issue(s) or topic(s) in depth in a structured way
Draw	draw a line to match a term with a description
Evaluate	judge or calculate the quality, importance, amount, or value of something
Examine	investigate closely, in detail
Explain	set out purposes or reasons / make the relationships between things evident / provide why and/or how and support with relevant evidence
Give	produce an answer from a given source or recall/memory
Identify	name/select/recognise
Justify	support a case with evidence/argument
Outline	set out main points
Predict	suggest what may happen based on available information
Sketch	make a simple freehand drawing showing the key features, taking care over proportions
State	express in clear terms
Suggest	apply knowledge and understanding to situations where there are a range of valid responses in order to make proposals
Summarise	select and present the main points, without detail
Write	write an answer in a specific way

Section 5: Example candidate response

This section takes you through an example question and candidate response. It will help you to see how to identify the command words within questions and to understand what is required in your response. Understanding the questions will help you to know what you need to do with your knowledge. For example, you might need to state something, calculate something, find something or show something.

All information and advice in this section is specific to the example question and response being demonstrated. It should give you an idea of how your responses might be viewed by an examiner but it is not a list of what to do in all questions. In your own examination, you will need to pay careful attention to what each question is asking you to do.

This section is structured as follows:



Paper 3 – Question 2

Complete – add information to an incomplete diagram.

2 The TCP/IP protocol suite can be viewed as a stack with **four** layers.

(a) Complete the stack by inserting the names of the **three** missing layers.

Application layer

[3]

(b) BitTorrent is a protocol used at the Application layer for the exchange of data.

(i) State the network model used with this protocol.

State – write the answer clearly and succinctly, you many only need to use a few words.

(ii) State the use of BitTorrent.

(iii) Explain how applications use BitTorrent to exchange data.

Explain how – set out the method used to perform the given task showing that you understand how the method used is applied to the situation set out in the question.

(c) State **two** other protocols that are used at the Application layer for the exchange of data.

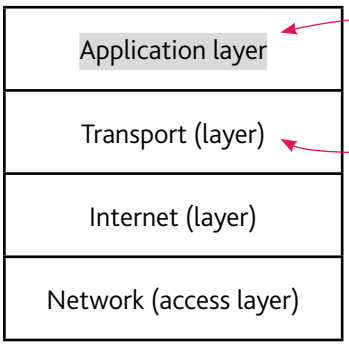
For each protocol, give a different example of an appropriate exchange of data.

Give – recall from your own knowledge.

This question requires knowledge and understanding of the TCP / IP stack, together with knowledge of protocols and understanding of what they are used for and how they are used.

Mark scheme – Paper 3 – Question 2

(a)



Answer already given on paper greyed out as not allowed

Content in brackets not essential

Both the name and the correct position on the diagram are needed

One mark for each correct layer in the correct place

(b) (i) Peer-to-peer

(ii) File sharing

Only correct answers here, no alternatives

(iii) Any **four** points from:

You only need four of these points for full marks.

BitTorrent client software made available

A computer joins a swarm by using this to load a Torrent descriptor file

A server called a tracker that keeps records of all the computers in the swarm …

.., and shares their IP addresses allowing them to connect to each other]

… at the beginning of an answer shows that this point is only worth a mark if the previous point has been included in the answer.

One computer in the swarm must have a complete copy of the torrent to be shared

Torrent is split into small pieces

Pieces of the torrent are both downloaded and uploaded]

Once a computer has a piece it can become a seed and upload

Leeches download much more than they upload

(c) Any two protocols from:

One mark for protocol and one mark for example

These answers are pairs, you must first identify the protocol to be awarded a mark for its use.

HTTP/HTTPS …

… Used for transfer of web pages from server to client

FTP …

… Used for interactive file transfer

SMTP …

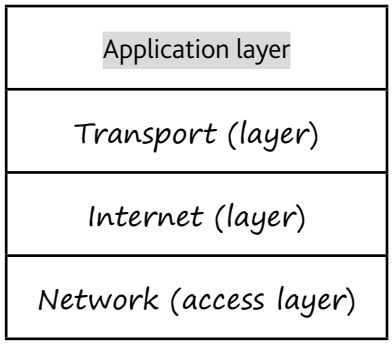
… Used for sending email messages

POP3/IMAP

… Used for retrieval of email messages

Now let's look at the example candidate response to the question and the examiner comments.

Example candidate response

Example Candidate Response – high		Examiner comments
(a)	 1	1 Each layer has been correctly identified. Mark for (a) = 3/3
(b)		
(i)	*Peer – to – peer* 2	2 The candidate has stated the correct protocol. Mark for (b)(i) = 1/1
(ii)	*File sharing* 3	3 The candidate has stated what BitTorrent is used for. There is no need to add more details as this is not required by the question. Mark for (b)(ii) = 1/1
(iii)	*BitTorrent client software is made available,* 4 *this is used to load the torrent descriptor for the required* 5 *file by computers joining it swarm. A server, called tracker, keeps records of all the computers joining the swarm and allows them to connect to each other by sharing their IP addresses.* 6 *The torrent is split into small pieces that can be downloaded or uploaded by each computer in the swarm. Once a* 7 *computer has downloaded a piece of the torrent file it can upload that piece to other computers in the swarm and become a seed.*	4 This part of the question requires an answer that explains how applications exchange data using BitTorrent. The candidate has described the software use. 5 The candidate states what the software is used for. 6 The candidate explains how the process is managed. 7 The candidate describes what is uploaded and downloaded. The final sentence is not required as the candidate has already gained all of the marks available, but it does complete the explanation and show that the candidate understands how the process is performed. Mark for (b)(iii) = 4/4
(c)	*Protocol 1 SMTP* 8 *Example Sending email messages* *Protocol POP3* *Example retirement of email messages*	8 The candidate has stated two correct protocols used at the application layer and examples are clear, correct and concise. No further description is required. Mark for (c) = 4/4 **Total mark awarded = 13 out of 13**

Common mistakes candidates made in this question

(a) Including the layer already given in the question or use layers from the OSI model.

(b) (ii) Describing what BitTorrent is used for e.g. streaming movies, this is not asked for in the question.

(iii) Use bullet points with single or very few words. The answer to part (iii) requires an explanation.

(c) Giving examples that are too vague e.g. email.

General advice

It is always a good idea to read the question carefully, noticing the command words and key instructions (in this case 'Complete', 'State', 'Give' and 'Explain how'). You may want to underline them to help you think about what they mean. Many candidates jump straight into writing their answer only to realise too late that they've answered a slightly different question to the one on the examination paper. Read the question first and pause to think about what you need to include before you start writing your answer – this will help you to choose an appropriate method of answering the question, so you don't waste time in the examination.

When answering a question that includes technical computer science terms, the example question includes several technical terms ('TCP / IP protocol suite', 'application layer', 'BitTorrent etc.'). It is often helpful to highlight the terms in the question. The terms and the command words used will help you to think about how you are going to answer the question.

Using correct computer science terminology in your answers will help you express your ideas clearly as well as making it easier for someone else to understand what you have done.

When answering a question that involves calculations, many candidates jump straight into writing their working only to realise too late that they've used the wrong method. Read the question first and pause to think about what you need to find before you start doing any working – this will help you to choose an efficient method, so you don't waste time in the examination. Don't forget that your working is part of your solution and you can gain method marks even if you don't get as far as a correct answer.

If you have had a good attempt at a question and still not managed to finish it, it is best to move on to another question and come back to it later. This will help you to make good use of the time you have available.

Allow a few minutes at the end of the examination to check your work. This will help you to spot errors in your answers that could lose you marks.

Section 6: Revision

This advice will help you revise and prepare for the examinations. It is divided into general advice and specific advice for each of the papers.

Use the tick boxes ☐ to keep a record of what you have done, what you plan to do or what you understand.

General advice

Before the examination

- ☐ Find out when the examinations are and plan your revision so you have enough time for each topic. A revision timetable will help you.
- ☐ Find out how long each paper is and how many questions you have to answer.
- ☐ Know the meaning of the command words used in questions and how to apply them to the information given. Highlight the command words in past papers and check what they mean. There is a list on page 11 of this guide.
- ☐ Make revision notes; try different styles of notes. See the *Learner Guide: Planning, Reflection and Revision* (www.cambridgeinternational.org/images/371937-learner-guide-planning-reflection-and-revision.pdf) which has ideas about note-taking. Discover what works best for you.
- ☐ Work for short periods then have a break. Revise small sections of the syllabus at a time.
- ☐ Build your confidence by practising questions on each of the topics.
- ☐ Make sure you practise lots of past examination questions so that you are familiar with the format of the examination papers. You could time yourself when doing a paper so that you know how quickly you need to work in the real examination.
- ☐ Look at mark schemes to help you understand how the marks are awarded for each question.
- ☐ Make sure you are familiar with the technical terminology that you need for this syllabus. Your teacher will be able to advise you on what is expected.

During the examination

- ☐ Read the instructions carefully and answer **all** the questions.
- ☐ Check the number of marks for each question or part question. This helps you to judge how long you should be spending on the response. You don't want to spend too long on some questions and then run out of time at the end.
- ☐ Do not leave out questions or parts of questions. Remember, no answer means no mark.
- ☐ You do not have to answer the questions in the order they are printed in the answer booklet. You may be able to do a later question more easily then come back to an earlier one for another try.
- ☐ Read each question very carefully. Misreading a question can cost you marks:
  - Identify the command words – you could underline or highlight them.
  - Identify the technical terms and perhaps underline or highlight them too.
  - Try to put the question into your own words to understand what it is really asking.
- ☐ Read all parts of a question before starting your answer. Think carefully about what is needed for each part. You will not need to repeat material.
- ☐ **Answer the question.** This is very important!
  - Use your knowledge and understanding.
  - Do not write everything you know about a topic. Only use the information you need to answer the question.
- ☐ Make sure that you have answered everything that a question asks. Sometimes one part requires two things, e.g. 'Calculate …' and 'Show your working.'. It is easy to concentrate on the first request and forget about the second one.

- ☐ Always show your working. Marks are usually awarded for using correct steps in the method even if you make a mistake somewhere.
- ☐ Don't cross out any working in a calculation until you have replaced it by trying again. Even if you know it's not correct you may still be able to get method marks. If you have made two or more attempts, make sure you cross out all except the one you want marked.
- ☐ Make sure all your numbers are clear, for example make sure your '1' doesn't look like a '7'.
- ☐ If you need to change a word or a number, it is better to cross out your work and rewrite it. Don't try to write over the top of your previous work as it will be difficult to read and you may not get the marks.
- ☐ Don't write any pseudocode answers in two columns in the examination. It is difficult for the examiners to read and follow your working.

Advice for Papers 1 and 3

- ☐ Always use the logic gate symbols from the syllabus when drawing logic circuits.
- ☐ Always use the opcodes given on the syllabus or shown on the examination paper when writing assembly language instructions.
- ☐ Try and use capital letters when writing assembly language opcodes, SQL or pseudocode commands so they can be clearly recognised as commands by the examiner.
- ☐ Where possible use SQL and pseudocode commands that are given in the syllabus, any other commands should be identified and explained.

Advice for Paper 2

- ☐ Try and use capital letters when writing pseudocode commands so they can be clearly recognised as commands by the examiner.
- ☐ Where possible use pseudocode commands that are given in the syllabus, any other commands should be identified and explained.
- ☐ Annotate pseudocode with comments.
- ☐ Fully label diagrams.

Advice for Paper 4

- ☐ Remember you will need to write and test programs in the examination.
- ☐ Only use one of the recommended programming languages for your answers:
  - Java
  - Visual Basic
  - Python
- ☐ Be able to use your chosen programming language in console mode.
- ☐ Get plenty of practice at debugging and testing programs using your chosen programming language.
- ☐ Where possible use the same programming language for all your answers.

Revision checklists

The tables below can be used as a revision checklist: **It doesn't contain all the detailed knowledge you need to know, just an overview.** For more detail see the syllabus and talk to your teacher.

The table headings are explained below:

Topic	You should be able to	R	A	G	Comments
Topics in the syllabus you need to cover.	Content in the syllabus you need to cover.	You can use the tick boxes to show when you have revised an item and how confident you feel about it. R = RED means you are really unsure and lack confidence; you might want to focus your revision here and possibly talk to your teacher for help A = AMBER means you are reasonably confident but need some extra practice G = GREEN means you are very confident As your revision progresses, you can concentrate on the RED and AMBER items in order to turn them into GREEN items. You might find it helpful to highlight each topic in red, orange or green to help you prioritise.			You can use the 'Comments' column to: • add more information about the details for each point • add formulae or notes • include a reference to a useful resource • highlight areas of difficulty or things that you need to talk to your teacher about or look up in a textbook.

Paper 1 – Theory Fundamentals

Topic	You should be able to	R	A	G	Comments
Data Representation	• Understand binary integers including, magnitudes, addition and subtraction and understand how overflow can occur	☐	☐	☐	
	• Convert between binary, denary, hexadecimal and Binary Coded Decimal (BCD)	☐	☐	☐	
	• Understand binary representation of character data including ASCII (American Standard Code for Information Interchange) and Unicode	☐	☐	☐	
Multimedia	• Understand bitmapped and vector image encoding and justify their use	☐	☐	☐	
	• Estimate bitmap image sizes	☐	☐	☐	
	• Understand representation of sound and the impact of changing the sampling rate and resolution	☐	☐	☐	
Compression	• Understand the need for and use methods (lossy and lossless) of compression for text, bitmap, vector graphics and sound files	☐	☐	☐	
Communication	• Understand networking including LANs (local area networks), WANs (wide area networks), devices, models (client-server and peer-to-peer), wireless and wired communication, routers, ethernet	☐	☐	☐	
	• Understand the difference between the World Wide Web (WWW) and the Internet	☐	☐	☐	
	• Understand the use of IP addressing, Uniform Resoruce Locator (URL) and Domain Name Service (DNS)	☐	☐	☐	
Hardware	• Understand the need for and the operation of input, output and storage devices including embedded systems	☐	☐	☐	
	• Understand and use logic gates with a maximum of two inputs	☐	☐	☐	
	• Construct logic circuits, truth tables and logic expressions	☐	☐	☐	

Topic	You should be able to	R	A	G	Comments
Processor Fundamentals	• Understand the Von Neuman model for a computer system including the Fetch-Execute (F-E) cycle, interrupts and ports • Understand the role of an assembler, a simple assembly language program that includes different modes of addressing and bit manipulation	☐	☐	☐	
System Software	• Describe an Operating System (OS), the tasks performed, why an OS is needed, the role of utility programs and program libraries • Understand the need for language translators, Integrated Development Environments (IDE), the role of compilers and interpreters and their benefits and drawbacks	☐	☐	☐	
Security, privacy and data integrity	• Understand the difference between security, privacy and integrity of data • Describe security measures for stand-alone PCs and networks • Describe and use methods of data validation (on entry) and data verification (on entry and during transfer)	☐	☐	☐	
Ethics and Ownership	• Understand the need for ethical computer professionals, copyright legislation, software licensing and Artificial Intelligence (AI)	☐	☐	☐	
Databases	• Understand the limitations of a file-based approach • Understand and use relational database terminology, entity-relationship (E-R) diagrams, the normalisation process to 3NF and the features of a Database Management System (DBMS) • Understand and use simple Structured Query Language (SQL), Data Definition Language (DDL) commands and Data Manipulation Language (DML) commands	☐	☐	☐	

Paper 2 – Fundamental Problem-solving and Programming Skills

Topic	You should be able to	R	A	G	Comments
Computational thinking skills	• Understand the purpose and benefits of abstraction • Describe and use decomposition	☐	☐	☐	
Algorithm Design	• Write pseudocode from structured English or a flowchart • Describe and use stepwise refinement	☐	☐	☐	
Data Types and structures	• Select and use basic data types including records • Select and use 1D and 2D arrays • Understand and use text files • Understand and use Abstract Data Types (ADTs) including stacks, queues and linked lists • Write pseudocode for 1D and 2D arrays and to process array data	☐	☐	☐	

Programming	• Write pseudocode statements for:
	  • the declaration of variables and constants
	  • the assignment of values to variables and constants
	  • the expressions involving any of the arithmetic or logical operators input from the keyboard and output to the console
	  • IF, CASE and loop statements.
	• Define and use procedures and functions
		☐	☐	☐	

Topic	You should be able to	R	A	G	Comments
Software Development	• Understand the purpose, principles, benefits, drawbacks and the need for different development cycles	☐	☐	☐	
	• Understand the stages of the program development life cycle	☐	☐	☐	
	• Use structure charts to decompose a problem	☐	☐	☐	
	• Use state-transition diagrams to document an algorithm	☐	☐	☐	
	• Test and maintain a program including error identification and location, test strategy and planning, choice of test data, different types of maintenance	☐	☐	☐	

Paper 3 – Advanced Theory

Topic	You should be able to	R	A	G	Comments
Data Representation	• Understand and use user-defined data types	☐	☐	☐	
	• Understand methods of file organisation and access including use of hashing algorithms	☐	☐	☐	
	• Understand and use normalised floating point numbers including conversion to and from denary	☐	☐	☐	
Communication and internet technologies	• Understand communication protocols and the TCP / IP protocol suite	☐	☐	☐	
	• Understand circuit switching and packet switching	☐	☐	☐	
Hardware	• Understand Reduced Instruction Set Computer (RISC) and Complex Instruction Set Computer (CISC) processors, the use of pipelining, and registers, the four basic computer architectures, the characteristics of massively parallel computers and the concept of a virtual machine	☐	☐	☐	
	• Produce truth tables for logic circuits including full and half adders	☐	☐	☐	
	• Understand and use Boolean algebra and Karnaugh maps (K-map)	☐	☐	☐	

Topic	You should be able to	R	A	G	Comments
System Software	• Understand how an Operating System (OS) can maximise the use of resources, manage processes and provide a user interface • Understand virtual memory, paging and segmentation • Understand interpretation, stages of compilation, Backus-Naur Form (BNF) and Reverse Polish Notation (RPN)	☐	☐	☐	
Security	• Understand how encryption (symmetric and asymmetric) works • Understand a digital certificate • Be aware of Secure Socket Layer (SSL) and Transport Layer Security (TLS) including the purpose, use and application	☐	☐	☐	
Artificial Intelligence (AI)	• Understand the use of graphs, neural networks and machine learning for AI	☐	☐	☐	
Algorithms	• Understand linear and binary searching, bubble and insertion sorts, Abstract Data Types (ADTs) (stack, queue, linked list and binary tree), use of recursion, file and exception handling • Understand the efficiency of algorithms using Big O notation • Understand programming paradigms including low-level, imperative, object oriented and declarative	☐	☐	☐	

Paper 4 – Practical

Topic	You should be able to	R	A	G	Comments
Programming	• Write programs for linear and binary searching, bubble and insertion sorts, using ADTs (stack, queue, linked list and binary tree) of recursion, file and exception handling • Write imperative (procedural) and object-oriented programs	☐	☐	☐	

Section 7: Useful websites

The websites listed below are reliable useful resources to help you study for your Cambridge International AS and A Level Computer Science.

Programming tutorials including SQL	www.w3schools.com/
Program development	
java	www.jetbrains.com/idea/documentation/
	www.jetbrains.com/idea/download/#section=windows -
Visual Basic	https://visualstudio.microsoft.com/vs/express/
Python	www.python.org/downloads/
Declarative programming	
Prolog	www.swi-prolog.org/
Technical terms	
British Computer Society Glossary	www.bcs.org/category/5656
Revision	www.teach-ict.com/

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