SAMPLE MATERIAL

We are working with Cambridge Assessment International Education to gain endorsement for this forthcoming title.

Working for over 25 YEARS WITH Cambridge Assessment International Education

Cambridge Assessment International Education

Endorsed for full syllabus coverage

Cambridge IGCSE™ and O Level

Computer Science

Second edition

David Watson
Helen Williams

Boost

HODDER EDUCATION

Please note this is a sample and not a full chapter

We are working with Cambridge Assessment International Education to gain endorsement for this forthcoming series.

Written by renowned expert authors, our updated resources enable the learner to effectively navigate through the content of the updated Cambridge IGCSE and O Level Computer Science syllabuses (0478/0984/2210).

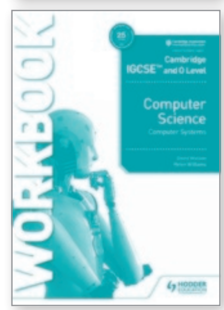
Cambridge IGCSE™ and O Level Computer Science Second Edition £27 March 2021 9781398318281	Cambridge IGCSE™ and O Level Computer Science Second Edition Boost eBook Boost £18 for 1-year access March 2021 9781398320765	Cambridge IGCSE™ and O Level Computer Science Computer Systems Workbook £8 March 2021 9781398318496	Cambridge IGCSE™ and O Level Computer Science Algorithms, Programming and Logic Workbook £8 March 2021 9781398318472	Cambridge IGCSE™ and O Level Computer Science Teacher's Guide with Boost Subscription Boost £125 for access until 2026 March 2021 9781398318502

Boost eBooks – interactive, engaging and completely flexible

Boost eBooks use the latest research and technologies to provide the very best learning experience for students. They can be downloaded onto any device and used in the classroom, at home or on the move.

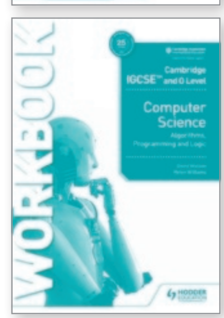
Develop understanding of computer systems, the internet and emerging technologies with further practice questions and activities

- The workbook provides a series of questions designed to test and develop knowledge of how computer systems and associated technologies work.



Develop algorithmic and computational thinking and programming skills with further practice questions and activities

- The workbook provides a series of questions designed to test and develop computational thinking skills in order to solve problems.



Created with teachers and students in schools across the globe, Boost is the next generation in digital learning for schools and colleges, bringing quality content and new technology together in one interactive website

The Teacher's Guide includes a print handbook and a subscription to Boost, where you will find a range of online resources to support your teaching.

- Confidently deliver the revised syllabus with a scheme of work, answers for all Student's Book activities and sample exam-style questions with annotated sample answers.
- Aid understanding with vocabulary flashcards and audio to aid pronunciation and offer support to ESL learners.

To explore the entire series,
visit **www.hoddereducation.com/cambridge-igcse-computerscience**

We're here to help!
To find out more, please contact us at **international.sales@hoddereducation.com**

Cambridge IGCSE™ and O Level

Computer Science

Second Edition

David Watson
Helen Williams

Boost
HODDER EDUCATION
AN HACHETTE UK COMPANY

Contents

Orders: please contact Bookpoint Ltd, 130 Park Drive, Milton Park, Abingdon, Oxon OX14 4SE. Telephone: +44 (0)1235 827827. Fax: +44 (0)1235 400401. Email education@bookpoint.co.uk Lines are open from 9 a.m. to 5 p.m., Monday to Saturday, with a 24-hour message answering service. You can also order through our website: www.hoddereducation.com

ISBN: 978 1 3983 1828 1

First published in 2010

This edition published in 2021 by
Hodder Education,
An Hachette UK Company
Carmelite House
50 Victoria Embankment
London EC4Y 0DZ

www.hoddereducation.com

Impression number 10 9 8 7 6 5 4 3 2 1

Year 2025 2024 2023 2022 2021

Illustrations by Aptara Inc.

Typeset by Aptara Inc.

Printed in the UK

A catalogue record for this title is available from the British Library.

1 Data Representation

In this chapter you will learn about:

★ Number systems:
  - how and why computers use binary to represent data
  - the denary, binary and hexadecimal number systems
  - converting numbers between denary, binary and hexadecimal
  - how and why hexadecimal is used for data representation
  - how to add two positive 8-bit numbers
  - overflow when performing binary addition
  - logical binary shifts on positive 8-bit integers
  - two's complement notation to represent positive and negative binary numbers

★ Text, sound and images
  - how and why a computer represents text
  - the use of character sets including ASCII and Unicode
  - how and why a computer represents sound
  - sound sample rate and sample resolution
  - how and why a computer represents an image
  - the effects of the resolution and colour depth on images

★ Data storage and compression
  - how data storage is measured
  - calculating the file size of an image and sound file
  - the purpose of and need for data compression
  - lossy and lossless compression

This chapter considers the three key number systems used in computer science, namely binary, denary and hexadecimal. It also discusses how these number systems are used to measure the size of computer memories and storage devices, together with how sound and images can be represented digitally.

1.1 Number systems

1.1.1 Binary represents data

As you progress through this book you will begin to realise how complex computer systems really are. By the time you reach Chapter 10 you should have a better understanding of the fundamentals behind computers themselves and the software that controls them.

You will learn that any form of data needs to be converted into a binary format so that it can be processed by the computer.

However, no matter how complex the system, the basic building block in all computers is the binary number system. This system is chosen because it only consists of 1s and 0s. Since computers contain millions and millions of tiny 'switches', which must be in the ON or OFF position, they can be represented by the binary system. A switch in the ON position is represented by 1; a switch in the OFF position is represented by 0.

Switches used in a computer make use of logic gates (see Chapter 10) and data is stored in memory locations, known as registers.

1.1.2 Binary, denary and hexadecimal systems

The binary system

We are all familiar with the denary number system which counts in multiples of 10. This gives us the well-known headings of units, 10s, 100s, 1000s and so on:

(10^4) **10 000**	**(10^3)** **1000**	**(10^2)** **100**	**(10^1)** **10**	**(10^0)** **1**
2	5	1	7	7

Denary uses ten separate digits, 0-9, to represent all values. Denary is known as a base 10 number system.

The **binary number system** is a base 2 number system. It is based on the number 2. Thus, only the two 'values' 0 and 1 can be used in this system to represent all values. Using the same method as denary, this gives the headings 2^0, 2^1, 2^2, 2^3 and so on. The typical headings for a binary number with eight digits would be:

(2^7) **128**	**(2^6)** **64**	**(2^5)** **32**	**(2^4)** **16**	**(2^3)** **8**	**(2^2)** **4**	**(2^1)** **2**	**(2^0)** **1**
1	1	1	0	1	1	1	0

A typical binary number would be: 11101110.

Converting from binary to denary

It is fairly straightforward to change a binary number into a denary number. Each time a 1 appears in a column, the column value is added to the total. For example, the binary number (11101110) above has the following denary value:

$$128 + 64 + 32 + 8 + 4 + 2 = 238 \text{ (denary)}$$

The 0 values are simply ignored.

Activity 1.1

Convert the following binary numbers into denary:

a 0 0 1 1 0 0 1 1

b 0 1 1 1 1 1 1 1

c 1 0 0 1 1 0 0 1

d 0 1 1 1 0 1 0 0

e 1 1 1 1 1 1 1 1

f 0 0 0 0 1 1 1 1

g 1 0 0 0 1 1 1 1

h 1 0 1 1 0 0 1 1

i 0 1 1 1 0 0 0 0

j 1 1 1 0 1 1 1 0

Converting from denary to binary

The reverse operation, converting from denary to binary, is slightly more complex. There are two basic ways of doing this.

Method 1

Consider the conversion of the denary number, 107, into binary. This method involves placing 1s in the appropriate position so that the total adds up to 107:

128	64	32	16	8	4	2	1
0	1	1	0	1	0	1	1

Method 2

This method involves successive division by 2. The remainders are then read from BOTTOM to TOP to give the binary value. Again using 107, we get:

2	107		
2	53	remainder:	1
2	26	remainder:	1
2	13	remainder:	0
2	6	remainder:	1
2	3	remainder:	0
2	1	remainder:	1
2	0	remainder:	1
	0	remainder:	0

read the remainder from bottom to top to get the binary number:

0 1 1 0 1 0 1 1

Activity 1.2

Convert the following denary numbers into binary (using both methods):

a 41 **b** 67 **c** 86 **d** 100 **e** 111
f 127 **g** 144 **h** 189 **i** 200 **j** 255

The hexadecimal system

The **hexadecimal number system** is very closely related to the binary system. Hexadecimal (sometimes referred to as simply 'hex') is a base 16 system and therefore needs to use 16 different 'digits' to represent each value.

Because it is a system based on 16 different digits, the numbers 0 to 9 and the letters A to F are used to represent each hexadecimal (hex) digit. A in hex = 10 in denary, B = 11, C = 12, D = 13, E = 14 and F = 15.

Using the same method as for denary and binary, this gives the headings 16^0, 16^1, 16^2, 16^3 and so on. The typical headings for a hexadecimal number with five digits would be:

(16^4) 65 536	(16^3) 4096	(16^2) 255	(16^1) 16	(16^0) 1
2	1	F	3	A

A typical example of hex is 2 1 F 3 A.

Since 16 = 2^4 this means that FOUR binary digits are equivalent to each hexadecimal digit. The following table summarises the link between binary, hexadecimal and denary:

▼ **Table 1.1**

Binary value	Hexadecimal value	Denary value
0 0 0 0	0	0
0 0 0 1	1	1
0 0 1 0	2	2
0 0 1 1	3	3
0 1 0 0	4	4
0 1 0 1	5	5
0 1 1 0	6	6
0 1 1 1	7	7
1 0 0 0	8	8
1 0 0 1	9	9
1 0 1 0	A	10
1 0 1 1	B	11
1 1 0 0	C	12
1 1 0 1	D	13
1 1 1 0	E	14
1 1 1 1	F	15

Converting from binary to hexadecimal and from hexadecimal to binary

Converting from binary to hexadecimal is a fairly easy process. Starting from the right and moving left, split the binary number into groups of 4 bits. If the last group has less than 4 bits, then simply fill in with 0s from the left. Take each group of 4 bits and convert it into the equivalent hexadecimal digit using Table 1.1. Look at the following two examples to see how this works.

Example 1

1 0 1 1 1 1 1 0 0 0 0 1

First split this up into groups of 4 bits:

1 0 1 1	1 1 1 0	0 0 0 1

Then, using Table 1.1, find the equivalent hexadecimal digits:

B	E	1

Example 2

1 0 0 0 0 1 1 1 1 1 1 1 0 1

First split this up into groups of 4 bits:

1 0	0 0 0 1	1 1 1 1	1 1 0 1

The left group only contains 2 bits, so add in two 0s:

0 0 1 0	0 0 0 1	1 1 1 1	1 1 0 1

Now use Table 1.1 to find the equivalent hexadecimal digits:

2	1	F	D

Activity 1.3

Convert the following binary numbers into hexadecimal:

- **a** 1 1 0 0 0 0 1 1
- **b** 1 1 1 1 0 1 1 1
- **c** 1 0 0 1 1 1 1 1 1 1
- **d** 1 0 0 1 1 1 0 1 1 1 0
- **e** 0 0 0 1 1 1 1 0 0 0 0 1
- **f** 1 0 0 0 1 0 0 1 1 1 1 0
- **g** 0 0 1 0 0 1 1 1 1 1 1 1 0
- **h** 0 1 1 1 0 1 0 0 1 1 1 0 0
- **i** 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1
- **j** 0 0 1 1 0 0 1 1 1 1 0 1 0 1 1 1 0

Converting from hexadecimal to binary is also very straightforward. Using the data in Table 1.1, simply take each hexadecimal digit and write down the 4-bit code which corresponds to the digit.

Example 3

4	5	A

Using Table 1.1, find the 4 bit code for each digit:

0 1 0 0	0 1 0 1	1 0 1 0

Put the groups together to form the binary number:

0 1 0 0 0 1 0 1 1 0 1 0

Example 4

B	F	0	8

Again just use Table 1.1:

1 0 1 1	1 1 1 1	0 0 0 0	1 0 0 0

Then put all the digits together:

1 0 1 1 1 1 1 1 0 0 0 0 1 0 0 0

Activity 1.4

Convert the following hexadecimal numbers into binary:

a 6 C

b 5 9

c A A

d A 0 0

e 4 0 E

f B A 6

g 9 C C

h 4 0 A A

i D A 4 7

j 1 A B 0

Converting from hexadecimal to denary and from denary to hexadecimal

To convert a hexadecimal number to denary is fairly straightforward. Take each hexadecimal digit and multiply it by its value. Add the totals together to obtain the denary value.

Example 1

4	5	A

First write out the number using headings:

(16^2) 256	(16^1) 16	(16^0) 1
4	**5**	**A**

Then multiply each digit by its value:

256	16	1	
$(4 \times 256 = 1024)$	$(5 \times 16 = 80)$	$(10 \times 1 = 10)$	(NOTE: A = 10)

Finally, add all the totals together:

denary number = 1 1 1 4

Advice

You do not need to remember the manufacturer MAC ID numbers.

Find out more

Try to find the MAC addresses of some of your own devices (e.g. mobile phone and tablet) and those found in the school.

Link

Refer to Chapter 3 for more detail on MAC addresses.

Link

Refer to Chapter 3 for more detail on IP addresses.

Find out more

Try and find the IPv4 and IPv6 addresses of some of your own devices (e.g. mobile phone and tablet) and those found in the school.

Media Access Control (MAC) addresses

Media Access Control (MAC) address refers to a number which uniquely identifies a device on a network. The MAC address refers to the network interface card (NIC) which is part of the device. The MAC address is rarely changed so that a particular device can always be identified no matter where it is.

A MAC address is usually made up of 48 bits which are shown as 6 groups of two hexadecimal digits (although 64-bit addresses also exist):

NN – NN – NN – DD – DD – DD

or

NN:NN:NN:DD:DD:DD

where the first half (NN – NN – NN) is the identity number of the manufacturer of the device and the second half (DD – DD – DD) is the serial number of the device. For example:

00 – 1C – B3 – 4F – 25 – FE is the MAC address of a device produced by the Apple Corporation (code: 001CB3) with a serial number of: 4F25FE. Very often lower case hexadecimal letters are used in the MAC address: 00-1c-b3-4f-25-fe. Other manufacturer identification numbers include:

00 – 14 – 22 which identifies devices made by Dell

00 – 40 – 96 which identifies devices made by Cisco

00 – a0 – c9 which identifies devices made by Intel, and so on.

Internet Protocol (IP) addresses

Each device, when logging onto the internet, is given a ***unique*** address known as the **Internet Protocol (IP) address**. An IPv4 address is a 32-bit number written in denary or hexadecimal form: e.g. 109.108.158.1 (or 77.76.9e.01 in hex). IPv4 has recently been improved upon by the adoption of IPv6. An IPv6 address is a 128-bit number broken down into 16-bit chunks, represented by a hexadecimal number. For example:

a8fb:7a88:fff0:0fff:3d21:2085:66fb:f0fa

Note IPv6 uses a colon (:) rather than a decimal point (.) as used in IPv4.

Activity 1.5

1 Using software on your computer, for example, text colour option in *Word*, find out what colours would be represented by the following RGB denary value combinations:

	a		b		c
Red	53	Red	201	Red	112
Green	55	Green	122	Green	111
Blue	139	Blue	204	Blue	81

2 Convert each of the above denary numbers into hexadecimal.

EXTENSION

For those students considering A Level, the following section gives some insight into further study on encryption. This can be found in Topic 1 of the Cambridge International A Level syllabus (9618).

The following two exercises are designed to help students thinking of furthering their study in Computer Science at A Level standard. The two topics here are not covered in the IGCSE exam, and merely show how some of the topics in this chapter can be extended to this next level. The two topics extend uses of the binary number system and using two's complement format to do binary addition.

Topic 1: Binary Coded Decimal (BCD)

The **Binary Coded Decimal (BCD)** system uses a 4-bit code to represent each denary digit, i.e.:

0 0 0 0 = 0	0 1 0 1 = 5
0 0 0 1 = 1	0 1 1 0 = 6
0 0 1 0 = 2	0 1 1 1 = 7
0 0 1 1 = 3	1 0 0 0 = 8
0 1 0 0 = 4	1 0 0 1 = 9

Therefore the denary number, 3 1 6 5, would be 0 0 1 1 0 0 0 1 0 1 1 0 0 1 0 1 in BCD format.

Uses of BCD

The most obvious use of BCD is in the representation of digits on a calculator or clock display. Each denary digit will have a BCD equivalent value which makes it easy to convert from computer output to denary display.

Questions to try

1 Convert the following denary numbers into BCD format:
   a 2 7 1
   b 5 0 0 6
   c 7 9 9 0
2 Convert the following BCD numbers into denary numbers:
   a 1 0 0 1 0 0 1 1 0 1 1 1
   b 0 1 1 1 0 1 1 1 0 1 1 0 0 0 1 0

At the end of this chapter, you will have learned how to:
- ✔ use the binary and hexadecimal number systems
- ✔ convert numbers between the binary, denary and hexadecimal numbers systems
- ✔ add together two binary numbers
- ✔ carry out a logical shift
- ✔ store negative binary numbers using two's complement
- ✔ interpret ASCII and Unicode character tables
- ✔ understand the way a computer stores image and sound files
- ✔ represent the size of a computer memory using KiB, GiB and so on
- ✔ calculate the size of an image and sound file taking into account a number of factors
- ✔ understand the effect of sampling rates and resolution on the size of a sound file
- ✔ understand the effect of resolution and colour depth on the size of an image file
- ✔ understand the advantages and disadvantages of reducing the size of a file
- ✔ apply lossless and lossy file reduction techniques

Key terms used throughout this chapter

Binary number system – this is a number system based on 2 and can only use the values 0 and 1

Hexadecimal number system – a number system based on the value 16 (uses denary digits 0 to 9 and letters A to F)

MAC address – ***Media Access Control*** is an address that refers to a number which uniquely identifies a device on a network; it takes the form:

NN-NN-NN-DD-DD-DD

IP address – ***Internet Protocol*** identified either as IPv4 or IPv6; it gives a unique address to each device connected to the internet identifying their location

End of chapter exam-style questions

1 A software developer is using a microphone to collect various sounds for his new game. He is also using a sound editing app.
When collecting sounds, the software developer can decide on the sampling resolution he wishes to use.

a **i** What is meant by sampling resolution? [1 mark]

ii Describe how sampling resolution will affect how accurate the stored digitised sound will be. [3 marks]

The software developer will include images in his new game.

b **i** Explain the term image resolution. [1 mark]

ii The software developer is using 16-colour bitmap images. How many bits would be used to encode data for **one** pixel of his image? [1 mark]

iii One of his images is 16 384 pixels wide and 512 pixels high. He decides to save it as a 256-colour bitmap image. Calculate the size of the image file in gibibytes. [3 marks]

iv Describe any file compression techniques the developer may use. [3 marks]

2 The editor of a movie is finalising the music score. She will send the final version of her score to the movie producer by email attachment.

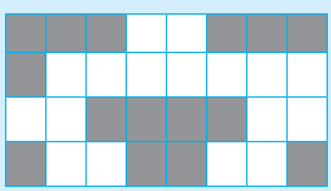
a Describe how sampling is used to record the music sound clips. [3 marks]

b The music sound clips need to undergo some form of data compression before the music editor can send them via email. Which type of compression, lossy or lossless, should she use. Give a justification for your answer. [3 marks]

c One method of data compression is known as run length encoding (RLE).

i What is meant by RLE? [3 marks]

ii The following image is being developed:



Show how RLE would be used to produce a compressed file for the above image. Write down the data you would expect to see in the RLE compressed format (you may assume that the grey squares have a code value of 85 and the white squares have a code value of 255). [4 marks]

3 a Convert the following denary numbers into 8-bit binary numbers:
i 123 **ii** 55 **iii** 180 [3 marks]

b Carry out the following additions using your binary values from **part (a)**:
i 123 + 55 **ii** 123 + 180 [4 marks]

c i Write down the two's complement value of: 0 1 1 1 0 1 0 0 [2 marks]

ii Write down the binary value of -112 using two's complement notation [1 mark]

iii Write down the denary value of the following binary number, which is using two's complement notation:

1	0	1	1	1	0	0	1

[1 mark]

d i Convert the following denary number into an 8-bit binary number using two's complement notation: 104

ii Use two's complement notation to find the 8-bit binary value of -104. [2 marks]

4 A bitmap image has the following resolution: 1140 x 1080 pixels. The image uses a colour depth of 24 bits.

a Explain the term **pixel**. [1 mark]

b Explain the term **colour depth**. [1 mark]

c Calculate how many of these images could be stored on a 32GiB memory stick. [3 marks]

d Describe how it would be possible to increase the number of these images which could be stored on this memory stick. [3 marks]

5 A memory stick is advertised as having a capacity of 64 GiB.

a How many photographs of size 10 KiB could be stored on this memory stick? [2 marks]

b John wants to store 400 photographs in a folder on his Solid State Drive (SSD). Each photograph is 10 KiB in size.

i Name **one** way of reducing the size of this file. [1 mark]

ii Give **two** advantages of reducing the size of his photography files. [2 marks]

iii Give **one** disadvantage of reducing files using the method named in **part (b)(i)**. [1 mark]

c The original photographs were stored as bitmap images.

i Explain why 3 bytes of data would be needed to store each pixel in the bitmap image. [2 marks]

ii Calculate how many different pixel colours could be formed if one of the bytes gives the intensity of the red colour, one of the bytes gives the intensity of the green colour and one of the bytes gives the intensity of the blue colour? [3 marks]

Written by renowned expert authors, our updated resources enable the learner to effectively navigate through the content of the updated Cambridge IGCSE and O Level Computer Science syllabuses (0478/0984/2210).

- **Develop computational thinking and problem-solving skills:** clearly-explained concepts are followed by opportunities to implement in the programming language of choice.
- **Build an understanding of computer systems and associated technologies:** carefully prepared worked examples explain new ideas alongside activities to test and consolidate.
- **Navigate the syllabus confidently:** supplementary subject content is flagged clearly, with introductions to each topic outlining the learning objectives.
- **Satisfy curiosity:** students are encouraged to deepen their knowledge and understanding of the subject with Extension activities and Find out more.
- **Consolidate skills and check understanding:** self-assessment questions, activities and exam-style questions are embedded throughout the book, alongside key definitions of technical terms.

This resource is endorsed by Cambridge Assessment International Education

- ✓ Supports the full Cambridge IGCSE™ and O Level Computer Science syllabuses (0478/0984/2210) for examination from 2023
- ✓ Has passed Cambridge International's rigorous quality-assurance process
- ✓ Developed by subject experts
- ✓ For Cambridge schools worldwide

We are working with Cambridge Assessment International Education to gain endorsement for this forthcoming series.

For over 25 years we have been trusted by Cambridge schools around the world to provide quality support for teaching and learning. For this reason we have been selected by Cambridge Assessment International Education as an official publisher of endorsed material for their syllabuses.

HODDER EDUCATION
e: education@bookpoint.co.uk
w: hoddereducation.com

ISBN 978-1-3983-1922-6