SAMPLE PAGES
International Computing for Lower Secondary

Deliver an exciting computing course for ages 11-14. The course covers the requirements of the national curriculum for England and is mapped to the Level 2 CSTA K-12 Computer Science Standards and the Cambridge Assessment International Education Digital Literacy Framework for Stages 7-9.

Available in the series:

£17.99
International Computing for Lower Secondary Student’s Book Stage 7 9781510481985
International Computing for Lower Secondary Student’s Book Stage 8 9781510481992
International Computing for Lower Secondary Student’s Book Stage 9 9781510482005

Our Student’s Books are also available in two digital formats:
The Student eTextbooks are downloadable version of the printed textbooks. They enable students to study on the move and provide a cost-effective way to resource your classroom.

The Whiteboard eTextbooks are an online, interactive version of the printed textbooks that are ideal for front-of-class teaching and lesson planning.

1 year access: £12.00/ 2 year access: £17.99
International Computing for Lower Secondary Student’s Book Stage 7 Student eTextbook 9781510483538
International Computing for Lower Secondary Student’s Book Stage 8 Student eTextbook 9781510483569
International Computing for Lower Secondary Student’s Book Stage 9 Student eTextbook 9781510483606

£50 for access until 31st Aug 2025
International Computing for Lower Secondary Student’s Book Stage 7 Whiteboard eTextbook 9781510483545
International Computing for Lower Secondary Student’s Book Stage 8 Whiteboard eTextbook 9781510483552
International Computing for Lower Secondary Student’s Book Stage 9 Whiteboard eTextbook 9781510483590

Online Teacher’s Guide
Save time preparing your lessons with a cost-effective range of online resources such as teaching notes, copyright guidance, curriculum links, glossary, 'how to use' instructions, Student Book answers, online safety advice, software list, CPD videos and more.

£125 for access until 31st Aug 2025
International Computing for Lower Secondary Online Teacher’s Guide Stage 7 9781510483484
International Computing for Lower Secondary Online Teacher’s Guide Stage 8 9781510483491
International Computing for Lower Secondary Online Teacher’s Guide Stage 9 9781510483507

To learn more about the series visit hoddereducation.com/international-computing
The Publishers would like to thank the following for permission to reproduce copyright material.

Photo credits

[TK]

Acknowledgements

[copy editor to add]

Every effort has been made to trace all copyright holders, but if any have been inadvertently overlooked, the Publishers will be pleased to make the necessary arrangements at the first opportunity.

Although every effort has been made to ensure that website addresses are correct at time of going to press, Hodder Education cannot be held responsible for the content of any website mentioned in this book. It is sometimes possible to find a relocated web page by typing in the address of the home page for a website in the URL window of your browser.

Hachette UK’s policy is to use papers that are natural, renewable and recyclable products and made from wood grown in well-managed forests and other controlled sources. The logging and manufacturing processes are expected to conform to the environmental regulations of the country of origin.

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: 9781510481992
© Margaret Debbadi and Siobhan Matthewson 2020

First published in 2020

This edition published in 2020 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 2024 2023 2022 2021 2020

All rights reserved. Apart from any use permitted under UK copyright law, no part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or held within any information storage and retrieval system, without permission in writing from the publisher or under licence from the Copyright Licensing Agency Limited. Further details of such licences (for reprographic reproduction) may be obtained from the Copyright Licensing Agency Limited, www.cla.co.uk

Cover photo [to come]

Illustrations by Aptara, Inc.

Typeset in FS Albert Regular 12/14pt by Aptara, Inc.

Printed in [to come]

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

8.1 Computer systems: The inside track 6
8.2 Networks and communication: Across the world in an instant 40
8.3 Html and CSS: Getting your head straight 68
8.4 High level programming language: Expert story telling 93
8.5 Spreadsheet modelling: Model my merch 114
8.6 Relational databases: SegwayThere 145

Glossary 171
Index 177
Introduction

About this book
This Student Book will continue to support your progression in computing and in the ongoing development of the skills needed to progress to specific curriculum areas such as IGCSE Computer Science and IGCSE ICT. Building on the skills developed in the previous book in this series you will further develop the technical skills needed to engage effectively in the digital world of today. It supports the curriculum areas of digital literacy, computer science and information technology:

KEYWORDS
Digital literacy focuses on the impact of digital technology in today’s society. It promotes an increased understanding of the impact of the digital world with an emphasis on maintaining safety and well-being online.
Computer science is the study of the inner workings of digital technology devices and the production of computer programs to solve problems. In computer science you will not only learn how to create computer programs to help computers solve problems, but also how the computer can interpret those instructions and carry them out.
Information technology looks at how to use previously developed computer programs such as web page development programs, movie making and presentation programs to solve problems for other people.
Units

This Student Book has six units:

8.1 Computer systems: The inside track introduces students to the inner workings of computer systems. It investigates the use of binary number representation and the use of logic gates for processing and decision making through a scenario around a space station holiday resort and robotic hosts that support guests during their stay.

8.2 Networks and communication: Across the world in an instant introduces the concept of networks and methods of electronic communication. The chapter uses the concept of an e-commerce application to investigate how organisations make effective use of electronic platforms to support world-wide communication.

8.3 HTML and CSS: Getting your head straight focuses on the development of web page content using html and CSS. In this chapter the focus is on the production of standardised web page presentations through the use of internal, inline and external CSS.

8.4 High level programming language: Expert story telling provides an introduction to high level programming through the development of an interactive adventure game using the programming language Python.

8.5 Spreadsheet modelling: Model my merch further develops skills in the use of spreadsheet applications. This chapter explores the use of more complex formula structures to support data analysis, decision making and the use of graphical presentations and. It also investigates how data input and presentation can be controlled using validation methods and cell formatting.

8.6 Relational database: SegwayThere encourages further understanding of the use of database applications by introducing the concept of relational databases to effectively store data. It provides a more in-depth look at controlling data input and ensuring effective management of data output using complex queries and report structures.
**How to use this book**

In each unit you will learn new skills by completing a series of tasks. Each unit starts with some information followed by a list of the learning objectives that you will cover. These features also appear in each unit:

- **SCENARIO**
  This panel contains a scenario which puts the tasks into a real-world context.

- **KEYWORDS**
  This panel lists important words on each page; the words are boldened the first time they appear in a unit. A definition is provided on the page. All keywords also feature in the glossary index at the back of the book.

- **Do you remember?**
  This panel lists the skills you should already be able to do before starting the unit.

- **Learn**
  This panel introduces new concepts and skills.

- **Practice**
  This panel contains tasks with step-by-step instructions to apply the new skills and or knowledge from the ‘Learn’ panel.

- **Computational Thinking**
  This panel highlights tasks in the unit which involve one of the key areas of computational thinking:

  - **Pattern recognition**: the identification of repeating tasks or features in a larger problem to help solve more complex problems more easily
  - **Decomposition**: breaking larger problems down into smaller more manageable tasks. Each smaller task is examined and solved more easily than a larger more complex problem.
  - **Abstraction**: ignoring details or elements of a problem which are not needed when trying to solve a problem.
  - **Algorithm development**: providing a series of instructions which include details on how to solve an identified problem
  - **Generalisation**: the process of creating solutions to new problems using past knowledge and experience to adapt existing algorithms.
  - **Evaluation**: the process of ensuring that an algorithmic solution is an effective and efficient one – that it is fit for purpose.
Introduction

**Go further**
This panel contains tasks to enhance and develop the skills previously learnt in the unit.

**Challenge yourself**
This panel provides challenging tasks with additional instructions to support new skills.

**Final project**
This panel contains the final tasks of the unit which encompass the all skills developed. It can be used to support self/peer assessment and teacher assessment.

**Evaluation**
This panel provides guidance on how to evaluate and, if necessary, test the Final Project.

**DID YOU KNOW?**
This panel provides an interesting or important fact about the task or theme.
Robots and AI

Robots and Artificial Intelligence (AI) are used more and more to help us complete everyday tasks and leading future developments is human interactions with machines. For example:

<table>
<thead>
<tr>
<th>Robotics</th>
<th>Artificial intelligence</th>
</tr>
</thead>
<tbody>
<tr>
<td>A humanoid robot called Atlas, created by Boston Dynamics, combines AI and robotics to tackle urban obstacle courses called ‘parkour’.</td>
<td>Voice-activated assistants such as Siri and Alexa use machine learning and AI to help provide high quality interactions with human users.</td>
</tr>
<tr>
<td>‘Soft robotics’ is the next generation of robots; they will be made of soft materials with a wide range of motion which can more closely mimic human movement.</td>
<td>Many online shopping applications use AI to track users’ internet searches to target advertising. They also make suggestions for purchase when the user is next online.</td>
</tr>
</tbody>
</table>

In this unit you will learn:

- about the use and impact of robotics and artificial intelligence in the travel industry
- that data is stored and manipulated in binary format inside computing devices
- how to use binary representation to store positive numbers
- that alternative representations such as hexadecimal (hex) can be used to represent data (including colour representations)
- how to convert numbers from decimal to binary and hexadecimal number systems
- how to perform simple additions and multiplications using binary values
- how sound files are stored in computer systems.

**SCENARIO**

SpaceVaca is a new travel agent. It specialises in trips to a holiday resort on a space station. The company wants to offer holiday makers a real space adventure where robots are used to staff the resort and welcome guests.

Your challenge is to research how robots and artificial intelligence (AI) can be used to help humans run the hotel. You will consider the pros and cons of using technology in this way. Once you have identified potential areas where robots and AI can be used, you will design a ‘robo-host’ which will be used to perform tasks, such as welcome new guests to the hotel.
Before you can design the robo-host you will need to consider how data is stored, manipulated and transferred around a computer system.

**Do you remember?**

Before starting this unit, you should be able to:

- ✔ add two numbers together
- ✔ multiply two numbers together
- ✔ use a spreadsheet to enter basic numbers and formula
- ✔ enter a URL into a web browser to access a specific web page
- ✔ list a range of input and output devices used with computers and other digital devices, for example using keyboards or microphones or touch screens for input, screens or speakers for output
- ✔ list a range of sensors used to provide inputs to computers and other digital devices, for example temperature sensors, humidity sensors, light sensors or motion sensors.

You should also know that:

- ✔ images can be represented in bitmap format
- ✔ you can use flowcharts to illustrate the steps involved in solving a complex problem
- ✔ data can be input and output using a range of devices including sensors
- ✔ sensors and input and output devices can be used to provide data and receive information from digital devices
- ✔ spreadsheets can be used to record data and carry out calculations using formulas.

**KEYWORD**

**bus:** a set of wires used to transfer data around the inside of a computer system

**DID YOU KNOW?**

The word ‘robot’ comes from a Czech term ‘robota’ which means ‘forced labour’ or ‘drudgery’. Robots are used in, for example:

- manufacturing (for example, when building cars)
- space exploration (for example, when unmanned robots collect samples from other planets)
- healthcare (for example, to help conduct surgery when the doctor and patient are in different locations)
Robotic holiday

When we think of robots we may think about how they are used in car assembly plants, space travel or even medicine. It’s unlikely we would think about using robots and AI in the travel industry. However, when we combine robotics with AI we can create a machine that mimics the way humans think. A robot can work out solutions to problems and learn from past experiences, as well as carry out the steps to complete a task. For these reasons, robots are increasingly being used to help travellers in the travel industry.

Hotels and the travel industry also use AI and - sometimes - we do not even realise we are interacting with a machine. Chatbots allow us to make bookings or ask questions online, even when hotel or travel agency staff are not available. Chatbots are now being used by some health care organisations to ensure patients have constant access to medical support.

In some countries, robots are used to help with airport security to detect concealed weapons. You may have even seen people walking through airport lounges being followed by their automated suitcase!

Using technology this way has its pros and cons. For example:

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Robots do not get tired, they can work 24 hours a day, seven days a week.</td>
<td>● High cost of development and maintenance</td>
</tr>
<tr>
<td>● Many tasks are completed more quickly by robots.</td>
<td>● Complex programs needed for their operation.</td>
</tr>
<tr>
<td>● Tasks are completed to the same standard at all times.</td>
<td>● Need to be reprogrammed if task changes; cannot easily handle changes in tasks.</td>
</tr>
<tr>
<td>● Tasks are completed more quickly.</td>
<td>● May be prone to bugs.</td>
</tr>
<tr>
<td>● They do not need to be paid.</td>
<td>● Security concerns; can be hacked.</td>
</tr>
<tr>
<td>● Tasks are completed without error.</td>
<td>● Staff will need to be retrained.</td>
</tr>
<tr>
<td>● They can work in dangerous environments.</td>
<td>● Some staff may lose their jobs.</td>
</tr>
<tr>
<td>● Can be programmed to interact in many different languages.</td>
<td>● Robots are unable to express emotions and provide empathy in the same way humans can.</td>
</tr>
</tbody>
</table>
Practice

With a partner, discuss the possible ways robotic technology and AI could be used by SpaceVaca. Think about, and make notes on, the following:

➤ The point where the customer accesses the SpaceVaca website to make a booking.
➤ When the customer makes their booking and pays for their holiday.
➤ When the customer arrives at the hotel.
➤ During the customers stay at the hotel.
➤ When the customer checks out and leaves the hotel.

Produce a poster advertising SpaceVaca and the robotic and AI services it provides for its customers. Your poster can be digital or hard copy. It should include a range of graphics and make people want to visit the space station for a holiday.

Your poster should tell a story to show:
➤ how the customers will use robotic technology and AI for at least three tasks during their stay or when booking their holiday
➤ how the customer will benefit from using robotic technology and AI for each task
➤ a warning for each task about some possible difficulties or disadvantages of using robotic technology and AI to complete each task.

Tell your partner about an experience you had using a chatbot.

KEYWORDS
chatbot: a computer program designed to mimic a human conversation
empathy: the ability to understand the feelings of another person

You could think here about food, drink, payment, health care.
Collecting traveller input

Digital processing systems such as mobile phones, tablet computers or gaming stations are made up from a combination of electronic hardware devices (such as keyboards, microphones, touch screens, sensors) and software (or computer programs) which tell the hardware devices how to operate. For the system to be useful to the user, there must be a way of providing input (voice input or text input via a keyboard) to the system. This input is then processed using a computer program and the output is presented to the user.

<table>
<thead>
<tr>
<th>INPUT</th>
<th>PROCESS</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provided by the user in the form of data</td>
<td>Data is processed using a computer program</td>
<td>Results if processing are provided for the user</td>
</tr>
</tbody>
</table>

There are many ways users can provide input to and receive output from digital systems. Some of the most common input and output devices used with digital devices today are listed below.

<table>
<thead>
<tr>
<th>Input devices</th>
<th>Output devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Microphone</td>
<td>● Touch screen for output display</td>
</tr>
<tr>
<td>● Touch screen (menu options or keyboard entry)</td>
<td>● Speakers for sound output</td>
</tr>
<tr>
<td>● Sensors</td>
<td>● Printers</td>
</tr>
<tr>
<td>● Scanners (image scanners, biometric scanners)</td>
<td></td>
</tr>
<tr>
<td>● Trackpads (to move cursors on screen)</td>
<td></td>
</tr>
</tbody>
</table>

Users can provide instructions to computer systems in a number of ways.

➤ Users can enter commands from a limited list of instructions that the system recognises.
   • Commands could be selected from a menu using a cursor and a pointing device such as a mouse (or the user’s finger, as with the smart fridge example shown here).
   • Simple instructions could be typed into the device or selected using a keyboard, such as your television remote control or a touch screen keyboard on a tablet or a gaming console.
Users can use their own **natural language** (the language spoken by humans, using full sentences and phrases without breaking down instructions into key words or abbreviations) to enter commands into the computer system.

- Instructions could be spoken into the device using a microphone; for example, the way we issue instructions to Siri or Alexa.
- Instructions could be typed into the device using a keyboard using natural language. For example, we can type full sentences into internet search engines to find relevant content.

**KEYWORDS**

- **smart fridge**: a fridge with internal cameras and sensors which senses what products are stored inside and allows users to keep track of contents, using internet connections they can suggest potential recipes for users based on the contents of their fridge
- **digital processing system**: any system that processes digital data (that is, data in the form of 0s and 1s)
- **natural language**: the language spoken by humans; for example English, Chinese, Vietnamese, Japanese, Spanish

**Practice**

- With a partner, discuss the range of ways you can use your mobile phone. Consider the ways you can provide input and receive output from your mobile phone.
- When you have finished your discussion, your teacher will give you a copy of a diagram for you to complete to illustrate your findings.
- A smart fridge can be used to help with keeping track of food in the household, ordering shopping, and planning recipes. Discuss with your partner how AI and robotics could be used in future homes. Design a robot which can be used in the future to help you complete everyday tasks.
- Draw a picture of your robot and write a short description to explain:
  - the task it is designed to help you complete
  - the input and output devices it will use.
Complex programs which use AI are often needed in systems where users can enter instructions using natural language. For example, when we use voice-activated instructions with our mobile phones, we tend to speak in full sentences.

But how do these programs work out what the user’s instruction is? Look at the diagram below; we can see what happens when a user speaks an instruction into a digital device such as Google Assistant on a mobile phone.

- Instructions are converted to text and the key words identified.
- The key words can then be looked up in a list of possible commands.
- The instructions relating to those key words are carried out.
- Output is produced.
Practice

➤ With a partner, create a table similar to the one below with a list of digital devices you use each day. Beside each device, identify possible uses and the input and output devices for each use. Decide if you enter commands using natural language or from a fixed list of commands when completing that task. An example has been given for you.

<table>
<thead>
<tr>
<th>Device</th>
<th>Use</th>
<th>Input device</th>
<th>Output Device</th>
<th>Natural language or Fixed command set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile phone</td>
<td>Searching the internet</td>
<td>Touch screen keyboard</td>
<td>On screen display</td>
<td>Natural language</td>
</tr>
</tbody>
</table>

Game developers are looking at how voice activation can be used to control video games. Modern remote controls, such as those provided with Apple television, and tools such as Amazon Alexa, accept voice entry from users. This allows users to carry out many tasks, such as:

- change television channels
- schedule recordings
- search for television shows
- order goods online
- search for information
- book a taxi

➤ Select one of the tasks listed above and recreate the diagram from the Learn box to include:
  - The input instructions a user might use to successfully complete the task.
  - The key words the computer would use to help it understand the task.
  - The steps the computer might take to complete the task.
  - The output the computer would provide to confirm completion of the task.
Now that we know how instructions can be entered into a computer, let us think about what it might look like on the inside.

Computers do not recognise and store letters, words or numbers in the same way we do. Every piece of data stored in a computer is actually stored as a number. Any letters, symbols, pictures, sounds or videos we store are all converted into numbers. The numbers used to store all of this data are made up from combinations of 1’s and 0’s; each 1 or 0 is known as a binary digit or bit. This is because digital devices are made from millions of tiny switches called transistors. Each switch can only have one of two values: 1 (ON) or 0 (OFF).

When data is stored inside a computer, the bits used to represent data can be grouped and processed together in larger units as shown in this diagram. (You might recognise some of the terms used in the diagram; we use them to help describe how large a file is and the storage capacity of storage devices.)

![Diagram showing storage units]

8 bits grouped together is called a byte and can be used in computers to represent a single character of text on a computer, e.g. the following combinations of 1’s and 0’s (01000001) is used to represent the letter ‘A’ on most computer systems.

**KEYWORDS**

- **transistors**: a tiny switch that can be activated by electrical signals. If the transistor is ON it represents 1, when it is OFF it represents 0
- **binary digit**: 0 or 1, the smallest unit of data represented by a computer
- **bit**: short for binary digit
- **byte**: a group of 8 bits, often used to represent a single character in a computer
**Using binary to store text**

**Learn**

If our digital devices are storing all of our data as 1’s and 0’s, then how do we use binary to represent the letters and characters which make up the words and phrases we enter into the computer?

The set of characters that can be represented on any device is known as its **character set**. The character set is made up from all of the letters, numbers, punctuation symbols and **special characters** represented on a device. Each character will be represented by a distinct set of 1’s and 0’s.

Most devices use a common code so that the characters can easily be transferred from one computer to another. One of the most commonly used codes is the **American Standard Code for Information Interchange (ASCII)** (this can be pronounced as ‘as-kee’).

ASCII is a 7 bit code used to represent all of the keys on a standard keyboard. The table below shows some of these 7 bit codes. However, as computers normally store information in 8 bits, an extra 0 is added to the beginning of the ASCII code to make it up to 8 bits. Each byte contains a unique combination of 0s and 1s to represent each character on a standard keyboard.

<table>
<thead>
<tr>
<th>ASCII character</th>
<th>Binary</th>
<th>ASCII character</th>
<th>Binary</th>
<th>ASCII character</th>
<th>Binary</th>
<th>ASCII character</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space</td>
<td>0100000</td>
<td>8</td>
<td>0111000</td>
<td>P</td>
<td>1010000</td>
<td>h</td>
<td>1101000</td>
</tr>
<tr>
<td>!</td>
<td>0100001</td>
<td>9</td>
<td>0111001</td>
<td>Q</td>
<td>1010001</td>
<td>i</td>
<td>1101001</td>
</tr>
<tr>
<td>“</td>
<td>0100010</td>
<td>:</td>
<td>0111100</td>
<td>S</td>
<td>1010010</td>
<td>j</td>
<td>1101010</td>
</tr>
<tr>
<td>#</td>
<td>0100011</td>
<td>&lt;</td>
<td>0111101</td>
<td>T</td>
<td>1010100</td>
<td>k</td>
<td>1101011</td>
</tr>
<tr>
<td>$</td>
<td>0100100</td>
<td>=</td>
<td>0111110</td>
<td>U</td>
<td>1010101</td>
<td>l</td>
<td>1101100</td>
</tr>
<tr>
<td>%</td>
<td>0100101</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>m</td>
<td>1101101</td>
</tr>
</tbody>
</table>

When a key is pressed on a keyboard the ASCII code for that character is generated using a group of eight electrical pulses. For example pressing R on a keyboard causes 8 transistor switches to be on or off in a particular pattern shown in the diagram.

Pressing ‘R’ on a keyboard will generate the 8-bit ASCII code 01010010 which can be processed by the computer.

![Diagram showing ASCII code for R]

Further examples: ‘H’ = 0100 1000 and ‘i’ = 0110 1001, so the word ‘Hi’ would be represented as the following pattern of 1’s and 0’s 0100 1000 0110 1001.
International Computing for Lower Secondary

**Practice**

Use the ASCII code table on the previous page to show how the following words would be represented in ASCII.

- school
- computer

**Pattern recognition**

Look at the ASCII code chart shown on the previous page. What patterns do you notice with the characters? Discuss these with a friend.

**Abstraction**

The combinations of binary values listed below all represent phrases or questions that some guests have spoken said to the robo-host. Using the ASCII table, translate the binary values into English for the manager of the space station.

- 0100 0100 0110 0101 0110 1100 0110 1100 0110 1111
- 0101 0111 0110 1000 0110 0001 0111 0100 0111 0100 0110 1001 0110 1101 0110 0101 0110 1001 0111 0011 0110 1001 0111 0100 0011 1111

**Decomposition**

A guest has just asked the robo-host the following two questions.

- Can I check in now?
- What is my room number?

Use the ASCII table above create the binary code the robo-host would use to represent these questions.

Use the process of decomposition to break down this problem.

- You may wish to look at one character at a time
- Does the same character appear twice? How can you speed up the process of translating the words?

If you have time, send some messages to your friend using the ASCII code.

**Keywords**

**character set:** The set of characters that can be represented on any device

**special characters:** Symbols and other characters (aside from standard letters, numbers and punctuation symbols) which can be created using a standard keyboard, such as #, -, %

**American Standard Code for Information Interchange (ASCII):** a standard code used to represent text on computers and other electronic devices

Decomposition

Think about how long it took you to translate these messages. Your computer is carrying out billions of these conversions per second!
Using binary to store numbers

When we enter numbers into a digital device we may need to perform calculations using those values. For example, using a spreadsheet application to add two numbers together.

When numerical values are used for calculations computers cannot use the ASCII code to carry out these calculations. Instead the value is stored in **binary number format**. The binary number system got its name because it uses two symbols ‘1’ and ‘0’.

**Decimal number system**

Before we think about how binary numbers are represented let us revise the **decimal number system**. The decimal number system is the formal name for the numbers that we use every day.

Every number we want to represent in the decimal number system is made up from the ten digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. When you were younger you used units, 10s, 100s, 1000s, and so on, to help you understand the value of numbers greater than 9. For example, when we refer to the number 2409, what we actually mean is:

<table>
<thead>
<tr>
<th>Thousands</th>
<th>1000</th>
<th>Hundreds</th>
<th>100</th>
<th>Tens</th>
<th>10</th>
<th>Units</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>= 2 × 1000</td>
<td>= 4 × 100</td>
<td>= 0 × 10</td>
<td>= 9 × 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>400</td>
<td>0</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

So, the number 2409 is actually calculated as 2000 + 400 + 0 + 9.

Decimal is also known as **base 10** due to the fact that it contains 10 distinct digits (0–9).
KEYWORDS

binary number format: numbers represented using 1s and 0s

decimal number system: number system using the digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

place value: the numerical value a digit has as a result of its position in a number; for example, the number 24\(_{10}\) actually represents \((2 \times 10) + (4 \times 1)\)

base 10: another term for decimal as it is based on 10 digits

Practice

Rewrite the following numbers using the table layout from the previous page to show how the place values of a digit can alter its overall value in a number. Your teacher will provide you with a copy of these tables in the document Place Value Revision Exercise.doc

➤ Nine thousand
➤ Nine hundred and nine
➤ Seven thousand nine hundred and nine
➤ Seventy Four
Binary number system

Learn

We have seen that binary has only two digits 0 and 1. For this reason, the place value column headings in binary increase in multiples of 2 and each column can only contain 0 or 1.

So, for example, here is a binary number: 10010110₂. The subscript lets us know the number is a binary number. Here is what this actually means:

| Each column heading value is worked out by multiplying the previous heading by 2 |
|----------------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 64      | 32      | 16       | 8        | 4        | 2        | 1        |
| ▼       | ▼       | ▼       | ▼       | ▼       | ▼       | ▼       |
| 128     | 64      | 32       | 16       | 8        | 4        | 2        | 1        |
| 1       | 0       | 1        | 0        | 1        | 1        | 0        |
| = 128 × 1 | = 64 × 0 | = 32 × 1 | = 16 × 1 | = 8 × 0 | = 4 × 0 | = 2 × 0 | = 1 × 0 |
| 128     | 0       | 32       | 16       | 0        | 4        | 2        | 0        |

1
the digit or bit to the left hand side of the number is known as the Most Significant Bit (MSB) as it has the largest place value.

1
the digit or bit to the right hand side of the number is known as the Least Significant Bit (LSB) as it has the smallest place value.

So, the number 10010110₂ represents 128 + 0 + 0 + 16 + 0 + 4 + 2 + 0 which is 150₁₀. We use a subscript 10 to let us know that this is a decimal value.

Now that we understand how the place values work, we can easily convert any binary number into decimal.

Here are two more examples:

**Example 1:** 10110000₂

<table>
<thead>
<tr>
<th>128</th>
<th>64</th>
<th>32</th>
<th>16</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>= 128 × 1</td>
<td>= 64 × 0</td>
<td>= 32 × 1</td>
<td>= 16 × 1</td>
<td>= 8 × 0</td>
<td>= 4 × 0</td>
<td>= 2 × 0</td>
<td>= 1 × 0</td>
</tr>
<tr>
<td>128</td>
<td>0</td>
<td>32</td>
<td>16</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

So, 10110000₂ = 128 + 32 + 16 = 176₁₀
Example 2: $00001111_2$

<table>
<thead>
<tr>
<th></th>
<th>128</th>
<th>64</th>
<th>32</th>
<th>16</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$= 128 \times 0$</td>
<td>$= 64 \times 0$</td>
<td>$= 32 \times 0$</td>
<td>$= 16 \times 0$</td>
<td>$= 8 \times 1$</td>
<td>$= 4 \times 1$</td>
<td>$= 2 \times 1$</td>
<td>$= 1 \times 1$</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

So, $00001111_2 = 8 + 4 + 2 + 1 = 15_{10}$

We still use 1 byte (or 8 bits) to represent $15_{10}$ in binary even though we do not really need the first four bits of this byte (which are set to zero). This is because computers normally store data in fixed length groups of 8 bits. In the past, computers could only process 8 bits at a time.

So, even though we only need 4 bits to store the number 15 we always represent binary numbers using all 8 bits, including all of the zeroes.

When we think about how the number 15 is stored in computer memory we could also think of it as being stored in the transistors as:

```
OFF OFF OFF OFF ON ON ON ON
```

Practice

Use the tables below to show how the following binary numbers would be represented in decimal format.

(a) $\begin{array}{cccccccc}
128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 0
\end{array}$

(b) $\begin{array}{cccccccc}
128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 0
\end{array}$

(c) $\begin{array}{cccccccc}
128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 0
\end{array}$

(d) $\begin{array}{cccccccc}
128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0
\end{array}$

KEYWORDS

Most Significant Bit (MSB): the digit to the left hand side of the number; the bit with the largest place value

Least Significant Bit (LSB): the digit to the right hand side of the number; the bit with the smallest place value

fixed length: having a set number of bits to represent a value; for example, numbers in binary are normally shown using 8 bits
The hotel manager cannot understand how decimal numbers can be stored in computer memory as 0’s and 1’s and asks you to explain this process. Create a spreadsheet that will help explain the process.

- Open a blank spreadsheet file and save it as Binary Conversion.
- Set up a spreadsheet similar to the one show below.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place Values</td>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Binary Digits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON/OFF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Processing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decimal Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Each cell in this row contains one bit. The 8 bits together can represent a binary number which is stored in computer memory.

This row shows the ON/OFF pattern for each transistor used to store the binary digit shown in the row above.

This row calculates the decimal value of each bit entered in row 2 by multiplying the bit stored by its place value in the number.

Calculates the number stored by adding together place values multiplied by either 1 or 0 depending on the bit stored in each memory location.

- In cell I:4, enter a formula that will correctly convert the binary digit entered in cell I:2 into its decimal equivalent.
- Replicate this formula across the row, from cell I:4 to B:4, to calculate the decimal values for all bits in the binary number entered.

```
=I2*2^3
```

- Add a formula to cell B:6 which will calculate the decimal value being stored.

```
=SUM(B2:B8)
```
Enter a set of 8 binary digits into the spreadsheet to test it works correctly; for example, we know that 00001111 should generate the decimal value 15.

We can use the ON/OFF row in the spreadsheet to see how this binary number would be stored by a computer using 8 transistors. Each transistor can be switched ON or OFF.

Enter the following formula into cell B:3:

Enter the following formula into cell B:3:

Replicate this formula across the spreadsheet from cell B:3 to I:3 to work out if each of the other ‘transistors’ would be ON or OFF.

Test your spreadsheet using the binary code for 27 (00011011).

Entering 00011011 should produce the following results.

Ask a friend to test your spreadsheet with binary number representations they have calculated themselves. Ask them to complete this table to show the output from your spreadsheet. Your teacher will provide you with a copy of a table to complete, with these headings:
Decimal to binary

Learn

Humans arriving at the space station hotel will communicate with the robo-host using the decimal number system. We know the robo-host only understands binary. Any numbers input into the robo-host will need to be converted from decimal into binary first.

The easiest way to convert a number from decimal format into binary is by using the ‘divide by 2’ method.

We have already shown how the number 176\textsubscript{10} would look in binary format. Let us look at how the ‘divide’ by 2 method does this conversion.

<table>
<thead>
<tr>
<th>Number</th>
<th>Division</th>
<th>Remainder</th>
</tr>
</thead>
<tbody>
<tr>
<td>176</td>
<td>176 ÷ 2 = 88</td>
<td>remainder 0</td>
</tr>
<tr>
<td>88</td>
<td>88 ÷ 2 = 44</td>
<td>remainder 0</td>
</tr>
<tr>
<td>44</td>
<td>44 ÷ 2 = 22</td>
<td>remainder 0</td>
</tr>
<tr>
<td>22</td>
<td>22 ÷ 2 = 11</td>
<td>remainder 0</td>
</tr>
<tr>
<td>11</td>
<td>11 ÷ 2 = 5</td>
<td>remainder 0</td>
</tr>
<tr>
<td>5</td>
<td>5 ÷ 2 = 2</td>
<td>remainder 1</td>
</tr>
<tr>
<td>2</td>
<td>2 ÷ 2 = 1</td>
<td>remainder 0</td>
</tr>
<tr>
<td>1</td>
<td>1 ÷ 2 = 0</td>
<td>remainder 1</td>
</tr>
</tbody>
</table>

For example, a guest will think of their room number as being the decimal number 176 but the robo-host will store this in binary format as 10110000\textsubscript{2}.

Algorithmic thinking and abstraction

A guest arriving at the space station hotel must provide the robo-host with their room number so their bags can be taken to their room.

Using the flowchart, right your task is to design an algorithm which can be used to convert decimal numbers provided by guests into binary format so the robo-host can understand them. Flowchart statements have been provided but two are unnecessary. Use the process of abstraction to remove the unnecessary steps and then complete the algorithm by placing the remaining steps in the correct position in the flowchart.
Practice

The robo-host has provided the hotel manager with some information about hotel guests but it has output the information in binary format. Help the hotel manager by converting binary numbers below so they contain the correct decimal value.

- There are $00010111_2$ guests booked into the restaurant tonight.
- To date, there are $10110100_2$ bookings for next year.
- There have been $0011011110_2$ queries about bookings for next year.
- There are $11001100_2$ staff working in the space hotel today.

The robo-host needs to record details of how many guests are staying on each floor on the space station. A staff member entered the following digital values using a touch screen. Convert the following decimal values into 8 bit binary values for the robo-host. Show your working out:

<table>
<thead>
<tr>
<th>Floor 1</th>
<th>Floor 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>32 guests</td>
<td>127 guests</td>
</tr>
<tr>
<td>Floor 3</td>
<td>Floor 4</td>
</tr>
<tr>
<td>54 guests</td>
<td>76 guests</td>
</tr>
</tbody>
</table>
Binary calculations

The robo-host will be used to complete a range of tasks for guests and staff during their stay at the space station hotel. For example, it could help the manager work out how many guests are staying in the hotel at any one time, or it could calculate a total bill for guests at the end of their stay.

It is quite easy to add binary numbers together - the hardest calculation you have to do is 1 + 1. We all know that 1 + 1 = 2. But how do we represent 2 in binary?

Remember, the number 2 is written in binary as 10₂.

```
<table>
<thead>
<tr>
<th>128</th>
<th>64</th>
<th>32</th>
<th>16</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
= 128 × 0 = 64 × 0 = 32 × 0 = 16 × 0 = 8 × 0 = 4 × 0 = 2 × 1 = 1 × 0
| 0   | 0   | 0   | 0   | 0   | 0   | 2   | 0   |
```

This is what we are doing when we add binary values:

```
In binary addition, when adding two numbers, you will come across four cases:
0 + 0 = 0
1 + 0 = 1
1 + 1 = 0 carry 1 (which is 2₁₀ or 1₁₀)
If the carry value is added to a column that is already adding 1 + 1 then you get:
1 + 1 + 1 = 1 carry 1 (which is 3₁₀ or 11₁₀)
```

Now look at some more complicated binary additions:

**Example 1:** 0₁₀₂ + 0₁₁₂

```
<table>
<thead>
<tr>
<th>Binary place values</th>
<th>Stage 1</th>
<th>Stage 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2 1</td>
<td>2 1</td>
</tr>
<tr>
<td>0 1</td>
<td>+ 0₁₁ 1</td>
<td>+ 0₁₁ 1</td>
</tr>
<tr>
<td>+ 0₁₁ 1</td>
<td></td>
<td>1 0</td>
</tr>
<tr>
<td>1 0 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Remember that binary only uses two digits 0 and 1 so the biggest number we could ever produce by adding two digits together is 2₁₀ or, in binary, 10₂.

```
Adding 1 + 1 gives us an answer of 0 and we carry the 1 across to the next column.
```

We then add the carry to any other values in the column (in this case there are none, so we write down 1 under the next place value heading.

Just like adding in decimal, we start on the right hand side and continue adding and carrying until all columns are complete.