CAMBRIDGE
UNIVERSITY PRESS

Computer Science

for Cambridge IGCSE™ & O Level

Digital Teacher's Resource

Cambridge Assessment International Education
Endorsed for teacher support

CAMBRIDGE
UNIVERSITY PRESS

Computer Science

for Cambridge IGCSE™ & O Level

TEACHER'S RESOURCE

Victoria Ellis

CAMBRIDGE
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

314–321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre, New Delhi – 110025, India

79 Anson Road, #06–04/06, Singapore 079906

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9781108824200

First published 2016
Second edition 2021

20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

Printed in Malaysia by Vivar Printing

A catalogue record for this publication is available from the British Library

ISBN 978-1-108-82420-0 Digital Teacher's Resource

ISBN 978-1-108-82421-7 Digital Teacher's Resource Access Card

The information in the 'About the examination' section is based on the Cambridge IGCSE™, IGCSE (9–1) & O Level Computer Science syllabuses (0478/0984/2210) syllabuses for examination from 2023. You should always refer to the appropriate syllabus document for the year of your students' examination to confirm the details and for more information. The syllabus document is available on the Cambridge International website at *www.cambridgeinternational.org*

Cover image Nerthuz/Getty Images

Introduction

Welcome to the second edition of our very popular Cambridge IGCSE™ and O Level Computer Science series.

This new series has been designed around extensive research interviews and lesson observations with teachers and learners around the world following the course. As a result of this research, some changes have been made to the new series, with the aim of solving and supporting your biggest classroom challenges and developing your learners' passion and excitement for thinking like and communicating with a computer!

To bring this new series to life, we have introduced a 'Computer science in context' feature to the coursebook. Each chapter is introduced with an interesting real-life example of the topic, and throughout the chapters there are smaller examples linking learning to life outside the classroom walls to help you to further engage your learners. We have also included some case studies and questions drawing upon examples in industry, to help learners contextualise their learning.

We have increased our focus on developing learners' programming skills, from tasks in the coursebook, updated programming books for each of the three syllabus recommended languages (Visual Basic, Python and now, Java) and new chapters in each of our resources supporting the new programming scenario challenge. We have also introduced a 'skills focus' feature in the coursebook and programming books that hones in on key skills such as computational thinking and mathematical skills.

As we develop new resources, we ensure that we are keeping up-to-date with best practice in pedagogies. For this series we have added new features to the coursebook, such as 'getting started' questions and activities to help you to evaluate learners' learning starting points and 'reflection' questions after certain tasks to encourage learners to take control of their own learning journey. We have developed our differentiated support in this new series, with a three-tier approach to our programming tasks in the coursebook and programming books, progressing from 'getting started'/'demo', to 'practice', to 'challenge', supporting all your learners' different needs. You can read more about this in the 'Approaches to learning and teaching' section of this digital teacher's resource.

Finally, we have updated this digital teacher's resource to make it as useful and relevant as possible to your day-to-day teaching needs. From teaching activity, assessment and homework ideas, to how to tackle common misconceptions in each topic, to a new feature developing your own teaching skills, we hope that this handy resource will inspire you, support you and save you much-needed time.

We hope that you enjoy using this series and that it helps you to continue to inspire and excite your learners about this vital and ever-changing subject. Please don't hesitate to get in touch if you have any questions for us, as your views are essential for us to keep producing resources that meet your classroom needs.

Gemma Coleman

Commissioning Editor for IGCSE and O Level Computer Science, Cambridge University Press

About the authors

Sarah Lawrey

Sarah Lawrey is an author and education consultant with a great love for Computer Science. She taught the subject for several years to learners aged 11 to 18 years. She has authored several resources and developed syllabuses with different countries around the world. She also tutors teachers globally online to help them realise their potential of being a wonderful educator.

Victoria Ellis

Victoria Ellis has a degree in Computer Science and taught Computer Science and ICT for ten years. She has also worked on developing qualifications and resources to support the qualifications. Victoria has contributed to a number of text books and resources for schools.

Chris Roffey

Chris Roffey is an experienced ICT and Computer Science teacher in the UK and has written the previous edition of the programming book for Python as well as Cambridge University Press' Coding Club series. He organises the UK Bebras Challenge and TCS Oxford Computing Challenge on behalf of Oxford University.

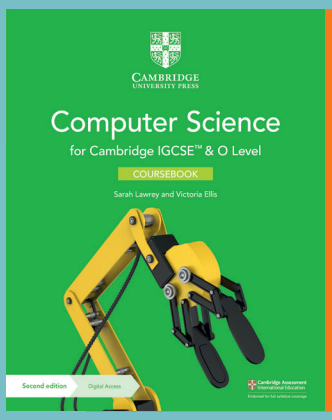
Richard Morgan

Richard Morgan has been teaching Computer Science in the UK for 16 years and is proud to say that many of the learners he has worked with have gone on to exciting and rewarding careers in the technology industry. Richard was born before the age of computers. As a young boy, Richard would read stories of wizards with unlimited access to knowledge and the power to communicate across worlds: the fantasy of his past is the reality of today with the use of mobile phones and the internet. He is excited by the challenges and rewards that the future of technology developments will bring.

Dave Duddell

Dave Duddell is a retired university lecturer. He has 20 years' experience working for examination boards as an examiner, moderator, team leader, reviser and trainer. He is a co-author of two editions of the Cambridge International AS and A Level Computer Science coursebook and two editions of the associated teacher's resources.

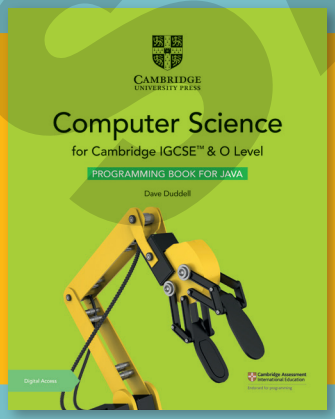
How to use this series



The coursebook provides coverage of the full Cambridge IGCSE™, IGCSE (9–1) and O Level Computer Science syllabuses (0478/0984/2210) for first examination from 2023. Each chapter explains facts and concepts and uses relevant real-world contexts to bring topics to life, including two case studies from Microsoft® Research. There is a skills focus feature containing worked examples and questions to develop learners' mathematical, computational thinking and programming skills, as well as a programming tasks feature to build learners' problem-solving skills. The programming tasks include 'getting started' skills development questions and 'challenge' tasks to ensure support is provided for every learner. Questions and exam-style questions in every chapter help learners to consolidate their understanding.

The digital teacher's resource contains detailed guidance for all topics of the syllabus, including common misconceptions to elicit the areas where learners might need extra support, as well as an engaging bank of lesson ideas for each syllabus topic. Differentiation is emphasised with advice for identification of different learner needs and suggestions of appropriate interventions to support and stretch learners.

The digital teacher's resource also contains scaffolded worksheets for each chapter, as well as practice exam-style papers. Answers are freely accessible to teachers on the 'supporting resources' area of the Cambridge GO platform.



There are three programming books: one for each of the recommended languages in the syllabuses – Python, Microsoft Visual Basic and Java. Each of the books are made up of programming tasks that follow a scaffolded approach to skills development. This allows learners to gradually progress through 'demo', 'practice' and 'challenge' tasks to ensure that every learner is supported. There is also a chapter dedicated to programming scenario tasks to provide support for this area of the syllabuses. The digital part of each book contains a comprehensive solutions chapter, giving step-by-step answers to the tasks in the book.

How to use this teacher's resource

This digital teacher's resource contains both general guidance and teaching notes that help you to deliver the content in our Cambridge IGCSE and O Level Computer Science coursebook. You will find **solutions** to the programming books, **answers** to the coursebook questions and exercises on the supporting resources area of Cambridge GO – they are freely available to teachers only.

There are **teaching notes** for each chapter of the coursebook. Please note that sometimes subsections in the coursebook are combined together in one sub-chapter of the digital teacher's resource. You can see an overview of where all topics are covered in the downloadable teaching plan, in the 'resources' column. Each set of teaching notes contains the following features to help you deliver the chapter.

At the start of each chapter there is a **teaching plan** for that chapter. This summarises the topics covered in the chapter, including the number of learning hours recommended for each topic, an outline of the learning content, and the resources from this series that can be used to deliver the topic:

Sub-chapter	Suggested learning hours	Learning content	Resources
6.1 Automated systems	3	Describe how sensors, microprocessors and actuators can be used in collaboration to create automated systems. Describe the advantages and disadvantages of an automated system in a given scenario.	**Coursebook:** 6.1 Automated systems **Worksheet 6.1:** Automated systems

This icon indicates that material is available to download from Cambridge GO (as part of this teacher's resource).

Each chapter also includes information on any **background knowledge** that learners should have before studying this chapter, advice on helpful **language support**, and a **teaching skills focus** that will help you develop your skills across a number of key pedagogical areas within the context of different syllabus topics.

BACKGROUND KNOWLEDGE

- Learners should be familiar with the function of AND, OR and NOT from their programming experience. They should also be aware that computers use logic gates through the hardware theory section.
- There is no other expected knowledge from learners.

Before the teaching notes for the individual sections, there are a selection of useful links to **digital resources**.

Links to digital resources

- Binary counter: interactive binary counter to see how the bits affect the number.
- RLE encoder: interactive text RLE encoder where text can be entered and it will identify repeated elements that can be compressed.

At the beginning of the teaching notes for the individual sections there is an outline of the **learning plan** for that section.

LEARNING PLAN

Syllabus learning objectives	Learning intentions	Success criteria
• Understand how and why computers use binary to represent all forms of data.	• Identify why computers represent data in binary.	Learners will be able to identify why computers represent data in binary.

There is also a feature highlighting any **common misconceptions** that learners may have about the topic and how you can overcome these.

Common misconceptions

Misconception	How to elicit	How to overcome
Learners may try and work out binary from left to right, i.e. 1 2 4 8, etc. instead of right to left, i.e. 8 4 2 1.	Ask learners to demonstrate how they convert binary to denary and vice-versa.	Correct learners when used incorrectly, and whenever giving examples place the values above the binary digits and reinforce right to left.

For each section, there is a selection of **starter ideas**, **main activities** and **plenary ideas**. You can pick out individual ideas and mix and match them depending on the needs of your class. The activities include suggestions for how they can be differentiated or used for assessment.

Starter ideas

1 Getting started (10 minutes)

Description and purpose: Ask learners to compare the words they have identified with another pair, and identify any words they have not included. Ask learners for their words and create a class-set on the board. Ask learners for definitions of their terms as they give them, and explain any concepts that come up.

At the end of the teaching notes for the individual sections are a selection **homework ideas**.

Homework ideas

1 Calculations

Description: Give learners a set of denary and binary numbers for them to convert to the other. This will allow learners to apply their knowledge.

This digital teacher's resource also includes **worksheets** for each chapter, a PowerPoint and two **practice exam-style papers**.

To help with lesson planning, a blank **lesson plan template** is available to download from Cambridge GO (as part of this digital teacher's resource).

How to use this teacher's resource to supplement PD

We regularly hear from teachers that the Professional Development (PD) they feel they get the most out of is face-to-face training. However, we also hear that not all teachers have the time or budget to get out of the classroom, so here's some handy suggestions and information about how to use this digital teacher's resource to supplement your own professional development. After all, we are all lifelong learners!

Teaching skills focus

We have created a new 'teaching skill focus' feature that appears once every chapter, covering a different teaching skill with suggestions of how you can implement it in the teaching of the topic. From differentiation, to assessment for learning, to metacognition, this feature aims to support you with trying out a new technique or approach in your classroom and reflecting upon your own practices.

Try it out once per teaching topic, or when you have time, and develop your skills in a supported and contextualised way.

Approaches to learning and teaching

Our teacher's resources now contain guidance on the key pedagogies underpinning our course content and how we understand and define them. You can find detailed information for you to read in your own time about active learning, assessment for learning, metacognition, differentiation, language awareness and skills for life taken from our *Approaches to learning and teaching* series.

Why not try reading each support document alongside the relevant Teaching Skills Focus for an extra bit of bedtime reading?

Teaching activity ideas

This digital teacher's resource contains a number of starter, main and plenary activity ideas for you to choose from and find the ones that work best in your classroom. Use them to support your creativity, breathe new life into a topic and build upon them with your own ideas.

The information in this section is based on the Cambridge IGCSE, IGCSE (9–1) and O Level Computer Science syllabuses (0478/0984/2210) for examination from 2023. You should always refer to the appropriate syllabus document for the year of your students' examination to confirm the details and for more information. The syllabus document is available on the Cambridge International website at www.cambridgeinternational.org.

About the syllabuses

The Cambridge IGCSE, IGCSE (9–1) and O Level Computer Science syllabuses (0478/0984/2210) have been updated for first examining from September 2023. They have been designed to provide a comprehensive and engaging introduction to Computer Science. The syllabuses are divided into ten topic areas. Component one of the syllabus on Computer Systems focuses on six topic areas: Data representation; Data transmission; Hardware; Software; The internet and its uses; and Automated and emerging technologies. Component two of the syllabus on Algorithms, Programming and Logic focuses on four topic areas: Algorithm design and problem-solving; Programming; Databases; and Boolean logic.

Learners are required to undertake practical work through the qualifications, gaining experience in decomposing problems, producing algorithms, writing computer programs in a high-level language, and exploring SQL in databases. These topics will be assessed on paper, so learners should also have experience of producing these algorithms without the use of a computer.

There are a number of changes in these qualifications from the previous versions such as:

- Introduction of new topics such as Artificial Intelligence and SQL.
- Reduction in some areas such as HTML.
- Expansion of programming requirements such as 2D arrays and file handling.
- Movement of topics between the papers such as Boolean logic.

Syllabus correlation grid from Cambridge GO.

About the examination

There are two papers: one is based on Computer Systems and the other one is is based on Algorithms, Programming and Logic. Learners need to take both papers.

Both papers cover the three Assessment Objectives.

The assessment objectives (AOs) are:

AO1

Demonstrate knowledge and understanding of the principles and concepts of computer science.

AO2

Apply knowledge and understanding of the principles and concepts of computer sciences to a given context, including the analysis and design of computational or programming problems.

AO3

Provide solutions to problems by:

- evaluating computer systems
- making reasoned judgements
- presenting conclusions.

The Computer Systems Paper consists of structured and free response questions covering topics 1–6 in the specification. This paper has a focus on computer science theory and applications of this theory. This paper has the largest focus on AO1 (60%) and then an equal split between AO2 and AO3. This means that there is more of an emphasis on knowledge and understanding of the topic areas, but there will be requirements to apply this knowledge and to make reasoned judgements.

The Paper on Algorithms, Programming and Logic consists of structured and free response questions covering topics 7–10 in the specification. This paper has a focus on problem solving with the 60% of the marks coming from AO2 – application of knowledge. There is then an equal spread between AO1 and AO3.

The final question in the Paper on Algorithms, Programming and Logic will be a banded response question. A scenario will be described that requires a computer program to solve, and learners will need to write an algorithm to solve this problem. This question is not marked by point, the mark scheme is banded and will list what is expected for each stage.

There is no choice of questions on any of the papers. They must all be answered.

Approaches to learning and teaching overview

The following are the key pedagogies underpinning our course content and how we understand and define them.

Active Learning

Active learning is a pedagogical practice that places student learning at its centre. It focuses on *how* learners learn, not just on *what* they learn. We, as teachers, need to encourage learners to 'think hard', rather than passively receive information. Active learning encourages learners to take responsibility for their learning and supports them in becoming independent and confident learners in school and beyond.

Active learning from Cambridge GO

Assessment for Learning

Assessment for Learning (AfL) is a teaching approach that generates feedback that can be used to improve learners' performance. learners become more involved in the learning process and, from this, gain confidence in what they are expected to learn and to what standard. We, as teachers, gain insights into a learner's level of understanding of a particular concept or topic, which helps to inform how we support their progression.

Assessment for Learning and **Improving learning through questioning** from Cambridge GO

Metacognition

Metacognition describes the processes involved when students plan, monitor, evaluate and make changes to their own learning behaviours. These processes help learners to think about their own learning more explicitly and ensure that they are able to meet a learning goal that they have identified themselves or that we, as teachers, have set.

Metacognition from Cambridge GO

Differentiation

Teachers need to find ways to welcome all learners and organise their teaching so that each student gets a learning experience that makes engagement and success possible. We should create a good match between what we teach and how we teach it, and what the learner needs and is capable of. We need not only to ensure access but also make sure each student receives the support and individual attention that result in meaningful learning.

Differentiation from Cambridge GO

Language awareness

For many learners, English is an additional language. It might be their second or perhaps their third language. Depending on the school context, students might be learning all or just some of their subjects through English.

For all students, regardless of whether they are learning through their first language or an additional language, language is a vehicle for learning. It is through language that students access the learning intentions of the lesson and communicate their ideas. It is our responsibility as teachers to ensure that language doesn't present a barrier to learning.

Language awareness and **Developing learners' language skills** from Cambridge GO

Skills for Life

How do we prepare students to succeed in a fast-changing world? To collaborate with people from around the globe? To use advanced thinking skills in the face of more complex challenges? At Cambridge we are responding to educators who have asked for a way to understand how all these different approaches to life skills and competencies relate to their teaching. We have grouped these skills into six main Areas of Competency that can be incorporated into teaching, and have examined the different stages of the learning journey, and how these competencies vary across each stage.

Skills for life from Cambridge GO

1 Data representation

Please note that sometimes multiple sub-chapters in the coursebook are combined together in one sub-chapter of this teacher's resource. Please check the 'resources' column of this teaching plan to see which coursebook sub-chapters are covered in each teacher's resource sub-chapter.

Teaching plan

Sub-chapter	Suggested learning hours	Learning content	Resources
1.1 Binary and hexadecimal	6	Understand why computers use binary. Convert between binary and denary, and vice-versa. Understand the use of hexadecimal. Convert between hexadecimal and denary, and hexadecimal and binary.	**Coursebook:** 1.1 Binary and hexadecimal **Programming Book for Python:** 8.2 FOR loops **Programming Book for Microsoft Visual Basic:** 5.3 Using the loop counter **Worksheet 1.1:** Hexadecimals
1.2 Binary manipulation and negative numbers	3	Add two positive 8-bit binary integers and understand the concept of overflow. Perform a logical binary shift left and right on a positive 8-bit binary integer. Use two's complement to represent negative 8-bit binary integers.	**Coursebook:** 1.2 Binary manipulation and negative numbers **Addition.pptx** **Worksheet 1.2:** Two's complement
1.3 Text, sound and images	3	Understand how and why a computer represents text, and the use of character sets, including ASCII and Unicode. Understand how a computer represents sound. Understand how a computer represents an image.	**Coursebook:** 1.3 How do computers represent text, images and sound? **Worksheet 1.3:** Crack the code
1.4 Data storage and compression	2	Understand how data storage is measured. Calculate the file size of an image and sound file. Understand the purpose and need for data compression. Understand how lossy and lossless compression works.	**Coursebook:** 1.4 Measuring data storage 1.5 Data compression

BACKGROUND KNOWLEDGE

- Learners should be aware that computers store data in binary. Some learners may have converted binary numbers into denary in mathematics.
- Learners should know that a computer can store a range of data including sound and video.
- Learners should be familiar with data storage values, most likely bytes, kilobytes, megabytes and gigabytes. They are unlikely to have used kibibytes, etc. and will need the differences clearly explaining.
- This section does not require learners to understand how the sound and images are captured, i.e. how a microphone works, they only need to be able to explain how the recorded data is stored in a file.

TEACHING SKILLS FOCUS

Area of focus: active learning. The focus of learning is on the learners themselves to support how they learn, instead of what they are learning. By exploration of a topic, and physical interaction, they are learning through their own successes and failures.

Specific focus: independent work. This is where learners explore a topic individually, without reliance on another person. It does not stop learners from interacting with peers, but they are performing the tasks on their own.

Benefits of independent work: supports research and exploration skills, requires learners to make decisions about what they do next, and allows them to make mistakes and find their own solutions.

Practice: give learners a grid where they can draw their own images by colouring in the squares. By selecting binary codes for the colours they have chosen they can explore how images are stored. By encouraging this exploration learners will develop independent skills.

The use of active learning allows learners to work out answers for themselves, by performing the role of the computer instead of just looking at how it works.

Learners will come across problems, for example, that 3 bits is not enough to represent 10 different colours. By actually finding and implementing the changes needed, they should remember this process more than if they were shown it.

Reflect: consider the following questions:

- Did the learners engage with the activity?
- Was there enough support for the less able to explore the topic?
- Were there enough extension activities to challenge the more able?
- How did learners tackle the problems they encountered? Were they reliant on being told the answer, or did they attempt to solve it themselves?
- What would you change next time you use active learning?

LANGUAGE SUPPORT

For definitions of key words, please see the glossary.

Some terms can appear in more than one place with a different use, for example, *resolution* can be *image resolution* or *sound resolution*. Make sure in discussions that the reference to image or sound is clear, i.e. not using resolution in isolation.

The data storage methods have very similar names and acronyms and learners need to be able to differentiate between these. The first three (*bit*, *nibble*, *byte*) can be related to eating a biscuit and the amount eaten. *Kibibyte* can be linked to kilo being 1000.

Lossy and lossless compression can be remembered by using their names, i.e. *lossy compression* means that data is lost permanently, whereas *lossless* means that all the data remains.

Links to digital resources

- Binary counter: interactive binary counter to see how the bits affect the number.
- RLE encoder: interactive text RLE encoder where text can be entered and it will identify repeated elements that can be compressed.
- Image storage: interactive image storage that lets you change the colour of a pixel and then shows how this changes the colour in the stored data.
- Bit shifting: interactive bit shifting webpage where learners can enter a binary number and then see the result of a shift.
- Binary calculator: interactive binary calculator that allows you to enter two binary numbers and outputs the addition.
- ASCII: the table of characters, numbers and symbols and their ASCII codes.
- UNICODE: the table of characters, numbers and symbols and their UNICODE codes.

1.1 Binary and hexadecimal

This section covers Coursebook sub-chapter topic 1.1.

LEARNING PLAN

Syllabus learning objectives	Learning intentions	Success criteria
• Understand how and why computers use binary to represent all forms of data.	• Identify why computers represent data in binary.	Learners will be able to identify why computers represent data in binary.
• Understand the denary, binary and hexadecimal number systems.	• Identify the number bases for denary and binary.	Learners will be able to identify that denary is base 10 and that binary is base 2.

• Convert between:
  • positive denary and positive binary
  • positive denary and positive hexadecimal
  • positive hexadecimal and positive binary.
	• Convert a binary number to denary.
	• Convert a denary number to binary.
	• Convert a binary number to hexadecimal.
	• Convert a hexadecimal number to binary.
	• Convert a denary number to hexadecimal.
	• Convert a hexadecimal number to denary.
		Learners will be able to convert between binary and denary.
		Learners will be able to convert between binary and hexadecimal, and denary and hexadecimal.

• Understand how and why hexadecimal is used as a beneficial method of data representation.	• Identify uses of hexadecimal. • Explain why hexadecimal is used.	Learners will be able to identify uses of hexadecimal and explain why hexadecimal is used.

Common misconceptions

Misconception	How to elicit	How to overcome
Learners may try and work out binary from left to right, i.e. 1 2 4 8, etc. instead of right to left, i.e. 8 4 2 1.	Ask learners to demonstrate how they convert binary to denary and vice-versa.	Correct learners when used incorrectly, and whenever giving examples place the values above the binary digits and reinforce right to left.
That the smallest binary number that can be represented is 0.	Ask learners what the smallest number is for a set number of bits.	Show learners that there is also the number 0.
The numbers above the binary digits start with 0 or 2, e.g. 8 4 2 or 8 4 2 1.	Ask learners to demonstrate how they convert binary to denary and vice-versa.	Correct learners where this is done and reinforce starting with 1 and then multiplying by 2 repeatedly.
Learners treat hexadecimal numbers such as 21 as the denary number 21.	Give learners example questions with these types of hexadecimal number.	Ask learners for their answers and make it clear when a number is hexadecimal and when it is not.
Learners mix the letters from 10 and above, e.g. allowing 16 as G.	Ask learners to write the hexadecimal values from 0 to F.	Show learners that only 0 to F can be represented in 4 bits.

Starter ideas

1 Getting started (10 minutes)

Description and purpose: Carry out the Getting started activity in Chapter 1 of the coursebook. Ask learners to compare the words they have identified with another pair, and identify any words they have not included. Ask learners for their words and create a class-set on the board. Ask learners for definitions of their terms as they give them, and explain any concepts that come up.

What to do next: Introduce the idea of data being stored in binary and that text, sound and images are all pieces of data that can be stored in binary.

2 Binary combinations (10 minutes)

Description and purpose: Show learners a 4-digit binary number, e.g. 0101. Ask learners to write as many different 4-digit binary numbers as they can think of. As learners finish tell them that there should be 16 different combinations, ask them to check if they have 16. Ask learners to swap their lists and identify which are missing (if any), and if there are any duplicates.

What to do next: Main teaching idea 1, 'Binary combinations'.

Main teaching ideas

1 Binary combinations (30 minutes)

Learning intention: Understand how to convert a binary number to denary.

Description and purpose: After Starter idea 2, ask learners to repeat the process with 5 binary numbers. Ask learners how many combinations they found and tell them that they should have 32. Ask learners to work out the relationship between the quantity of binary numbers and how many combinations there are. Link to number bases. Ask learners what standard decimal numbers are (denary), i.e. base 10. Explain that binary is base 2.

Differentiation ideas:

Support – give learners a paper template with columns for them to separate the binary digits and view them in a more structured manner.

Challenge – ask learners to look at 7+ digits and what the combinations are.

2 Conversion spreadsheet (30 minutes)

Learning intention: Convert denary numbers to binary and vice-versa.

Description and purpose: Recap basic spreadsheet formulae and how to implement these in a spreadsheet. Ask learners to work in pairs to create a spreadsheet that allows the user to enter an 8-bit binary number and calculate the value, and then the same for a denary number.

› **Differentiation ideas:**

Support – give learners a partially complete spreadsheet, e.g. where it tells them where the user will enter data, and tells them where to put their formulae.

Challenge – ask learners to increase up to 16-bits.

› **Assessment ideas:** Ask learners to test each other's systems and evaluate how well they work.

3 Hexadecimal and binary (20 minutes)

Learning intention: Understand how to convert a binary number to hexadecimal and vice-versa.

Resources: Worksheet 1.1, 'Hexadecimals'.

Description and purpose: Introduce learners to the process of converting a binary number into its hexadecimal equivalent; splitting into groups of 4, then giving each group one digit from 0 to F. Ask learners to complete Part 1 of the Hexadecimal worksheet. Ask learners for their answers and to demonstrate how they calculated them, give learners the correct answers for each question. Repeat but converting a hexadecimal number into binary by putting each digit into 4 bits. Ask learners to complete Part 2 of the Hexadecimal worksheet. Ask learners for their answers and to demonstrate how they completed them.

› **Differentiation ideas:**

Support – give learners a list of the binary numbers and the hexadecimal equivalents for them to identify the correct digits for each.

Challenge – ask learners to write a hexadecimal number for another learner to calculate.

› **Assessment ideas:** Ask learners to mark their own work and make corrections separately.

Plenary ideas

1 Special values (10 minutes)

Description and purpose: Ask learners to work out what is the smallest number that can be represented by 5 bits (0), the largest number that can be represented by 5 bits (1), and the quantity of different numbers that can be represented by 5 bits (32). Ask learners to repeat this with 6 bits. Ask learners to identify the pattern; the quantity of numbers is the next binary value, e.g. in 5 bits there are 32 combination which is the value of the 6th binary digit.

2 Number match (10 minutes)

Resources: List of binary, hexadecimal and denary numbers.

Description and purpose: Ask learners to match each binary number with its denary and/or hexadecimal number. These could be on a board, or on individual pieces of paper, of they could have to draw lines between them.

› **Assessment ideas:** Ask learners to check each other's answers and where there are differences work out what the correct answer is.

Homework ideas

1 Calculations

Description: Give learners a set of denary and binary numbers for them to convert to the other. This will allow learners to apply their knowledge.

2 Find your MAC/IP address

Description: Ask learners to find either the MAC address of one of their computers, or the IP address, and then write this number in denary, binary and hexadecimal. This will allow learners to apply their knowledge and explore future topics in networking.

1.2 Binary manipulation and negative numbers

This section covers Coursebook sub-chapter topic 1.2.

LEARNING PLAN

Syllabus learning objectives	Learning intentions	Success criteria
• Add two positive 8-bit binary integers.	• Perform binary addition on two positive 8-bit binary integers.	Learners will be able to add together two 8-bit binary integers.
• Understand the concept of overflow and why it occurs in binary addition.	• Explain when overflow occurs and its meaning.	Learners will be able to identify when overflow occurs and explain its meaning.
• Perform a logical binary shift on a positive 8-bit binary integer and understand the effect this has on the positive binary integer.	• Perform a left and right binary shift. • Explain the effect of a shift on a binary number.	Learners will be able to shift a binary number left and right, and explain the effect of a shift on a binary number.
• Use two's complement to represent negative 8-bit binary integers.	• Convert a positive binary integer to a two's complement 8-bit integer and vice-versa. • Convert a negative denary integer to a two's complement 8-bit integer and vice-versa.	Learners will be able to convert positive and negative denary integers into two's complement, and vice-versa.

Common misconceptions

Misconception	How to elicit	How to overcome
Learners may attempt to keep the bits lost in a left or right shift.	Ask learners to show on the board the result of a shift.	Give a narrative to the learners explaining if it's correct why – and if it's wrong why.
Learners may not carry in addition.	Give learners questions where they need to add binary numbers and show their working.	Work through an answer with the learner and reinforce how to use the carries.
Learners may attempt to add from left to right.	Give learners questions where they need to perform shifts.	Show learners how to work one answer and ask them to follow the same method for the next.
Learners may give a positive two's complement binary number starting with a 1.	Ask learners to write the two's complement for +200.	Show learners how leaving a 1 at the start will mean it's actually a negative number, and they have to add another 0 at the start. Link back to overflow in addition.

Starter ideas

1 Adding (10 minutes)

Resources: Addition.pptx.

Description and purpose: Show learners the first slide and ask them to convert the binary number to denary, and then add these values together. Ask learners for their answers and show them the answers. Ask learners to then convert these denary answers back into binary.

What to do next: Main teaching idea 1, 'Shifting'.

2 Shifting (10 minutes)

Resources: Document with binary numbers, as well as their shifted results.

Description and purpose: Give learners a list of binary numbers that have the same combinations of 1s and 0s but shifted, for example, 00001101 and 00011010. Ask learners to work out the difference between the two values. Repeat this with a number of questions. Ask learners for their answers and link to the role, e.g. multiplication or division by 2, 4, etc.

What to do next: Main teaching idea 3, 'Negative numbers'.

Main teaching ideas

1 Shifting (30 minutes)

Learning intention: Perform a logical left and right binary shift on a positive binary integer. Understand the loss of data in a logical shift.

Description and purpose: Explain that each shift to the left multiplies the binary number by 2. Show learners with an example, work out the denary of the number before and after.

Explain that each shift to the right divides the binary number by 2. Show learners with an example, work out the denary of the number before and after.

Show learners that the bits at either end of the register are lost when a logical shift is performed, e.g. 10101010 multiplied by 2 is 01010100. Discuss how logical shifts loses precision.

Write several binary numbers on a board and a shift for learners to perform on each one. Ask learners for their answers and to demonstrate the steps.

Differentiation ideas:

Support – put learners with a stronger learner when working through the answers.

Challenge – ask learners to consider how the drawback of loss of precision could be dealt with, such as using additional register to store the values.

2 Positive numbers (15 minutes)

Learning intention: Convert a positive binary integer to a two's complement 8-bit integer and vice-versa.

Resources: Worksheet 1.2, 'Two's complement'.

Description and purpose: Explain to learners that binary numbers can represent negative numbers. Show learners how the first bit in a two's complement binary integer identifies if it is positive (0) or negative (1). Explain how positive numbers are worked out exactly the same as ordinary binary numbers. Show learners a set of worked examples. Give learners Worksheet 1.2 and ask them to complete Part 1. Ask learners for their answers and to mark and correct their own work.

Differentiation ideas:

Support – give learners a binary table with the key values 8 4 2 1, etc. written in to help them.

3 Negative numbers (40 minutes)

Learning intention: Convert a negative binary integer to a two's complement 8-bit integer and vice-versa.

Resources: Worksheet 1.2, 'Two's complement'.

Description and purpose: Explain that when a two's complement binary number starts with a 1, it needs to be converted to work out its value. Select one method to demonstrate to learners how to convert a negative two's complement binary number to denary. Work through a number of negative

binary two's complement integers, ask learners to tell you what each stage is. Ask learners to complete Worksheet 1.2 Part 2. Ask learners for their answers and to mark and correct their own work.

Repeat, but turning a negative denary integer into two's complement. Ask learners to complete Worksheet 1.2 Part 3. Ask learners for their answers and to mark and correct their own work.

› **Differentiation ideas:**

Support – give learners a list of instructions to follow for each conversion that they can follow.

Challenge – ask learners to work out the smallest number that can be represented in 8-bits, the largest number and how many different numbers there are.

› **Assessment ideas:** Learners can mark and correct their own answers.

Plenary ideas

1 Shifting (10 minutes)

Resources: List of shifts and their results.

Description and purpose: Ask learners to match the shifts with their results such as 1 shift left will match with multiply by 2.

2 Two's complement match (10 minutes)

Resources: List of two's complement binary integers and their descriptions.

Description and purpose: Give learners a set of two's complement binary integers, and descriptions, e.g. the largest smallest 8-bit number, the two's complement binary integer −23, etc. and ask learners to match the descriptions with the answers. Ask learners for their answers and discuss any errors.

› **Assessment ideas:** Learners can check their own learning when reviewing the answers.

Homework ideas

1 Coursebook questions

Description: Ask learners to complete the questions in the coursebook and then their own answers in the next lesson. This will give learners experience of giving answers in writing.

2 Key binary and hexadecimal cards

Description: Give learners a set of cards with key words or questions from the Coursebook sub-chapter 1.2. Ask learners to complete the back with a description or definition. This will consolidate understanding of terms and provide a testing resource for future use.

1.3 Text, sound and images

This section covers Coursebook sub-chapter topic 1.3.

LEARNING PLAN

Syllabus learning objectives	Learning intentions	Success criteria
• Understand how and why a computer represents text and the use of character sets, including American standard code for information interchange (ASCII) and UNICODE.	• Explain how text is represented by a character set.	Learners will be able to describe how a computer represents characters, and the characteristics of ASCII and UNICODE.

CONTINUED

Syllabus learning objectives	Learning intentions	Success criteria
• Understand how and why a computer represents sound, including the effects of the sample rate and sample resolution.	• Describe how a sound wave is sampled to produce a sequence of binary values.	Learners will be able to define key terms such as sampling, sample rate, resolution, etc. Learners will be able to describe how a sound file is sampled and how its sample rate and sampling resolution will affect the sound and the sound file.
• Understand how and why a computer represents an image, including the effects of the resolution and colour depth.	• Describe how a bitmap image is stored in a computer, including key terms such as pixel, resolution and colour depth.	Learners will be able to describe how an image is represented in binary, and how the resolution and colour depth will affect the image and the image file.

Common misconceptions

Misconception	How to elicit	How to overcome
Learners often mix up the quality of sound and the similarity to the original.	Ask learners for the benefits of increasing the sampling resolution and sample rate.	Explain that sound does not have a quality, it is outputting a set sound. The overall sound file, e.g. when recording voices, sounds better because it is more like the original – the amplitudes are more like the original and there are few gaps between.
Learners usually explain that each colour has a code and miss that that code must be unique.	Ask learners how a bitmap image is stored in a computer.	Demonstrate a set of colours with codes – some of which are the same, and ask learners why it is important that the word unique is used.

Starter ideas

1 Crack the code (10 minutes)

Resources: Worksheet 1.3, 'Crack the code'.

Description and purpose: Give learners Worksheet 1.3 and ask them to decipher the message. Ask learners for the answer.

What to do next: Main teaching idea 1, 'Character sets'.

2 Image (15 minutes)

Resources: Print a 10 × 10 grid of squares for each learner and list of colours to use for each square.

Description and purpose: Ask learners to complete the grid by following the colour instructions given.

What to do next: Main teaching idea 3, 'Images'.

Main teaching ideas

1 Character sets (30 minutes)

Learning intention: Understand how and why a computer represents text using a character set.

Resources: ASCII and UNICODE character tables, e.g. ASCII and UNICODE.

Description and purpose: Explain that so far they have only looked at how numbers are represented. Introduce the idea of a character set and give learners a copy of a character set table. Ask learners to write a message using the character codes (denary or binary) – highlight the differences between capital (uppercase) and lowercase letters, and symbols including the space. Ask learners to swap their messages and try and work out what they wrote.

Differentiation ideas:

Support – give learners a limited character set with the denary/binary numbers and letters only (lowercase and uppercase), then slowly introduce more characters such as symbols.

Challenge – ask learners if they can find any patterns in the binary numbers, e.g. the value for B is 1 more than the value for b.

2 Sound (45 minutes)

Learning intention: Understand how a computer represents sound.

Resources: Example sound wave with x-axis labelled for milliseconds and y-axis labelled with amplitudes.

Description and purpose: Show learners the image of an analogue sound wave. Link to physics and how the height is the amplitude, the distance between the waves the frequency, etc. Explain that this is analogue data, i.e. it can be any value – but that computers only use binary. Show learners how the amplitude can be recorded and then stored as a binary number. Give learners the sound wave with amplitude values written on the axis, and ask them to write down the amplitude value in binary every millisecond. Ask learners to compare their answers with another learner and make relevant corrections. Extend this by asking learners to take samples more often, and/or less often and to identify the differences between the values recorded, i.e. more samples = more accurate to the original wave.

Differentiation ideas:

Support – give learners a table to complete with the milliseconds completed, for them to write the denary and then binary values alongside.

Challenge – ask learners to consider what will happen if the interval between samples is changed.

3 Images (30 minutes)

Learning intention: Understand how a computer represents an image, including the effects of the resolution and colour depth.

Resources: Grids to represent bitmap images.

Description and purpose: Ask learners to colour in the grid to make an image – each square must have only one colour in – with a maximum of 4 colours. Ask learners to write the name of each colour they have used and to give it a unique 2-bit binary number. Ask them to write the sequence of bits to represent the image they have drawn, e.g. the 2 bits for the first pixel, then the second, etc. Ask them to give their colour list and binary numbers to another learner to attempt to recreate the image. Link to key terms such as pixel, colour depth, etc. Ask learners questions such as what happens if you increase the number of bits per pixel? What will happen to the image (and the file size)?

Differentiation ideas:

Support – give learners a list of the colours they can use, and the binary values for each colour.

Challenge – ask learners to identify how many colours they could represent if they increased it to 3-bits per pixel, etc.

Plenary ideas

1 Write a message (10 minutes)

Resources: ASCII and UNICODE character tables, e.g. ASCII and UNICODE.

Description and purpose: Ask learners to write their name in the given character set, differentiating between capitals and lowercase letters.

2 Spot the difference (10 minutes)

Resources: Two strings that look similar but would not equate to the same, e.g. "my house" and "MyHouse".

Description and purpose: Tell learners that these two strings will not have the same values, e.g. if they were a password they would not match. Ask learners to identify all of the reasons why the two strings are not the same.

Homework ideas

1 Key images and sound cards

Description: Give learners a list of the key terms about text, images and sound. Ask them to write a definition on the back of each. This will allow learners to consolidate their understanding of terms and produce a future testing resource.

2 Exam-style questions

Description: Give learners a set of exam-style questions about text, images and sound. Review the answers in the next lesson. This will allow learners to experience the style of questions they will be required to answer in the examination.

1.4 Data storage and compression

This section covers Coursebook sub-chapter topics 1.4 and 1.5.

LEARNING PLAN

Syllabus learning objectives	Learning intentions	Success criteria
• Understand how data storage is measured.	• Understand that there are different data storage values.	Learners will be able to describe the different data storage values, and put these into numeric order.
• Calculate the file size of an image file and a sound file, using information given.	• Calculate the file size of an image file and a sound file.	Learners will be able to calculate the file size of an image file and a sound file.
• Understand the purpose of and need for data compression.	• Describe what is meant by compression and why it is needed.	Learners will be able to describe what is meant by compression and identify its purpose in specific contexts.
• Understand how files are compressed using lossy and lossless compression methods.	• Describe lossy and lossless compression and recommend one for a scenario.	Learners will be able to describe lossy and lossless compression. Learners will be able to select lossy or lossless compression for a given scenario and then justify their choice.

Common misconceptions

Misconception	How to elicit	How to overcome
Learners think that reducing the sound sampling rate reduces the quality of the sound produced. The quality of the sound is constant, the difference is that the sound is no longer identical to the original, i.e. it has changed.	Ask learners questions such as what happens if the sampling rate is increased or decreased?	Explain that sound is constant and its quality does not change. The change is that the value output is no longer the same as the original sound.

Starter ideas

1 Order the sizes (10 minutes)

Resources: List of different binary values, e.g. 3 GiB, 1 PiB.

Description and purpose: Give learners the list of binary values and ask them to put them in order from smallest to largest. Ask learners for their answers and reorder the values to show the learners the answer.

What to do next: Main teaching idea 2, 'Image file size'.

2 Sound and image terms (10 minutes)

Resources: List of descriptions of sound and image terms.

Description and purpose: Ask learners to identify the terms being described. Ask learners for their answers and review if they are correct and if not then why.

What to do next: Sound file size calculations.

Main teaching ideas

1 File sizes (20 minutes)

Learning intention: Understand how data storage is measured.

Description and purpose: Give learners a list of the acronyms and names for different sizes, e.g. GiB, bit. Ask learners to work in pairs to find out what they stand for and how large each file size is. Ask the pairs which is the smallest and then continue up to the largest, describing each one in turn.

› **Differentiation ideas:**

Support – give learners a document to complete with a table to write the full name and the size.

Challenge – ask learners to calculate how many bits are in each data storage, and if there are any values above exbibytes.

2 Image file size (30 minutes)

Learning intention: Calculate the file size of an image and sound file.

Resources: Descriptions of image files with their resolution and colour depth.

Description and purpose: Recap work on images and the features of an image file; resolution and colour depth. Ask learners how these values can be used to calculate the file size. Show learners, using an example, how to calculate the file size of an image. Give learners a set of questions that give learners the description of an image file and asks them to calculate the file size and present it in different values, e.g. how many bits, how many bytes, how many kibibytes?

› **Differentiation ideas:**

Support – structure the first few questions for learners, for example, having a formula where they fill in the gaps and then calculate.

Challenge – ask learners to find out what other data an image file might store other than the data for the pixels.

› **Assessment ideas:** Learners can mark their own answers and add corrections.

3 RLE (30 minutes)

Learning intention: Understand how lossless compression can compress a file.

Resources: Squared paper.

Description and purpose: Recap image work, e.g. pixels and colour depth. Ask learners to draw a bitmap image using the grid. Show learners how the image can be compressed using RLE by storing the colour and then the number of times it occurs in sequence for the whole image. Ask learners to give their compressed file data to another learner who has to decompress the data by redrawing the image.

Differentiation ideas:

Support – give the learner a structured document to help them encode their data, e.g. asking them to write the name of the first colour, and how many times it then occurs in a sequence.

Challenge – ask learners to consider how RLE can be used with other types of file, e.g. text or sound files.

Plenary ideas

1 File size match (10 minutes)

Resources: Descriptions of different files and their contents, and a list of file sizes.

Description and purpose: Ask learners to match the files with their file sizes. Go through each file and ask learners which file size it matches to, explain any differences.

2 Compression choice (10 minutes)

Resources: List of scenarios that need compression, e.g. an image file that is being sent by email to a friend.

Description and purpose: Ask learners to work in pairs to decide whether lossy of lossless compression should be used for each scenario, and to justify their answers. Ask the pairs for their answers and question why they have made that decision.

Homework ideas

1 Image vs text

Description: Give learners the description of an image and the thousand words statement, 'An image is worth a thousand words' – ask learners to use their understanding of files and file sizes to determine whether the statement is true or false. Learners should calculate the file size of the image, and the text file and then decide which is largest. This will allow learners to apply their understanding to another scenario, and then evaluate the result.

2 Exam questions

Description: Give learners a set of exam questions that require them to calculate file sizes, and/or that are about compression. Ask learners to answer the questions, then review these in the next lesson where learners can mark their own answers and make corrections where needed. This will allow learners to experience the style of questions they will need to answer.

2 Data transmission

Please note that sometimes multiple sub-chapters in the coursebook are combined together in one sub-chapter of this teacher's resource. Please check the 'resources' column of this teaching plan to see which coursebook sub-chapters are covered in each teacher's resource sub-chapter.

Teaching plan

Sub-chapter	Suggested learning hours	Learning content	Resources
2.1 Types and methods of data transmission	3	Understand that data are broken down into packets to be transmitted, including the structure of a packet. Describe how data is transmitted from one device to another using the following methods of data transmission: serial, parallel, simplex, half-duplex, full-duplex. Explain the suitability of each method of data transmission, for a given scenario. Understand the universal serial bus (USB) interface, explain how it is used to transmit data and evaluate its suitability for a given scenario.	**Coursebook:** 2.1 The structure of a data packet 2.2 Packet switching 2.3 Methods of data transmission 2.4 The universal serial bus (USB) interface **Worksheet 2.1:** Types of data transmission
2.2 Methods of error detection	2	Understand the need to check for errors after data transmission and how these errors can occur. Describe the processes involved in each of the following error detection methods for detecting errors in data after transmission: parity check (odd and even), checksum and echo check. Describe how a check digit is used to detect errors in data entry and identify examples of when a check digit is used, including ISBN numbers and bar codes. Describe how an automatic repeat request (ARQ) can be used to establish that data is received and that it is correct.	**Coursebook:** 2.5 Detecting errors in data transmission **Worksheet 2.2:** Parity
2.3 Encryption	2	Understand the need for encryption when transmitting data. Understand how data is encrypted using symmetric and asymmetric encryption.	**Coursebook:** 2.6 Encryption **Worksheet 2.3:** Encryption

BACKGROUND KNOWLEDGE

- Learners should be familiar with the idea of data being transmitted, for example, through the use of cables. They are not expected to have come across the different methods by which this data can be transmitted.
- Learners should be familiar with different types of cable, specifically those that make use of the USB interface.
- Learners should be aware that there can be errors in data transmission, but they are not expected to know why these occur, or the methods used to detect them.
- Learners should be familiar with the term encryption, and the use of encryption to avoid someone else understanding the original data. They may have come across some encryption routines, but this is not expected.

TEACHING SKILLS FOCUS

Area of focus: differentiation.

Specific focus: assessment for learning.

Benefits of assessment for learning: it allows learners to identify any missing gaps in knowledge, and their understanding. When used with exam-style questions learners can explore where their answers are insufficient to gain the marks, and what they will need to do in future exam-style questions.

Practice: develop a set of questions such as exam-style questions about data transmission. Either:

a increase the difficulty in the questions as they progress,

or

b create two or three different papers. For example, a less able, intermediate, and challenge set. Ask learners to select which set to take, or more than one if they want the option to change.

Once learners have answered the questions, work through each question with the class. Learners should be marking their own responses, adding corrections and making notes about what the questions and command words mean. These can be done in a different colour so that they can review the notes separately.

Once complete, ask learners to identify three goals based on the feedback; for example, apply each statement in a response to the context in the question.

Reflect: consider the following questions:

- Did learners select appropriate sets of questions to answer?
- Were learners supported but also challenged?
- Which learners required additional support to answer the questions?
- Were there any questions that learners could not answer?
- Were there any types of question that learners found more difficult?
- Would you use differentiation in this area again?
- What changes will you make next time?
- Did learners set themselves appropriate goals?
- Did learners gain a better understanding of a topic, or a style of exam question, from working through the answers?
- How will what you have learnt affect what you teach next?

LANGUAGE SUPPORT

For definitions of key words, please see the glossary.

The terms asymmetric and symmetric encryption can be used to indicate their function. *Symmetric* means the same, therefore the two devices have the same key. *Asymmetric* means not the same, so they have private and public keys.

The term *USB* is used commonly in a variety of means, but learners are expected to understand which of these are correct, and which are not. For example, a USB is not a device – USB is commonly talked about as being a memory stick but the device is not a USB, the memory stick uses a USB interface. Learners need to be clear about using this term correctly.

Links to digital resources

- Serial and parallel transmission: webpage explaining serial and parallel transmission.
- Parity: interactive system using an image to represent parity.
- Check digit calculator: online check digit calculator for a barcode.
- Encryption: video with facts about encryption.
- Symmetric encryption: webpage about symmetric encryption.
- Asymmetric and symmetric encryption: video about asymmetric and symmetric encryption.

2.1 Types and methods of data transmission

This section covers Coursebook sub-chapter topics 2.1–2.4.

LEARNING PLAN

Syllabus learning objectives	Learning intentions	Success criteria
• Understand that data is broken down into packets to be transmitted. • Describe the process of packet switching.	• Describe how a packet is broken down into parts. • Identify the different parts of a packet. • Describe how packets are transmitted and rearranged into order.	Learners will be able to describe how data is split into individual packets, that are then transmitted, and then rearranged by the recipient computer.
• Describe the structure of a packet	• Identify the parts of a packet and packet header.	Learners will be able to state the three parts of a packet; header, payload and trailer. State the purpose of each part of a packet. Identify the contents of the packet header, including the destination computer's address, packet number and originator's computer address.

CONTINUED

Syllabus learning objectives	Learning intentions	Success criteria
• Describe how data is transmitted from one device to another using different methods of data transmission.	• Describe each method of data transmission.	Learners will be able to describe how data is transmitted using serial or parallel. Learners will be able to describe how data is transmitted using simplex, half-duplex or full-duplex.
• Explain the suitability of each method of data transmission, for a given scenario.	• Identify and describe the use of the data transmission methods for a given scenario.	Learners will be able to identify whether a scenario should use serial or parallel, and simplex, half-duplex or full-duplex. Learners will be able to justify their choice.
• Understand the universal serial bus (USB) interface and, explain how it is used to transmit data.	• Describe the use of USB, how it transports data and whether it is appropriate for a given scenario.	Learners will be able to identify the key features of a USB interface. Learners will be able to describe the data transmission methods used by USB. Learners will be able to identify whether USB is appropriate, or inappropriate, for a given scenario, and justify their choice.

Common misconceptions

Misconception	How to elicit	How to overcome
That serial only has 1 wire.	Ask learners for the definition of serial.	Explain that there can be more than one wire because other data or signals are also sent, e.g. synchronisation signals. So there are multiple wires, but the data is only sent in one channel.
That data can only be one from serial, parallel, simplex, half-duplex or full-duplex. Or that full-duplex is always parallel, for example.	Ask learners to identify the two transmission types appropriate for a scenario.	Explain that serial and parallel are one option, and then simplex, half-duplex and full-duplex are a second option, and that there can be any combination from these 2 groups.
That USB is just a type of cable.	Ask learners to describe what is meant by USB.	Explain that the interface, i.e. the port and the method of communication as well as the wires within all make up the USB interface.
That USB is a data storage device.	Ask learners to describe what is meant by USB.	Explain that there are storage devices that make use of the USB interface, but they are not USB in themselves; i.e. memory stick, SSD.

Starter ideas

1 Getting started (10 minutes)

Description and purpose: Carry out the Getting started activity in Chapter 2 of the coursebook. Ask a pair of learners to act out their diagram, e.g. to write the data on paper and then explain how it is passed between them. Ask the other learners if they thought of the same method, or another. Ask each pair of learners who have a different method to describe it for the rest of the class. Create a list of the different methods and explain that they will be learning which of these is accurate or actually used.

What to do next: Identify the methods that are closest to how data can be transmitted. Begin with the different methods of transmission and keep referring back to the learners' original ideas.

2 Splitting a packet (10 minutes)

Description and purpose: Give each learner a different written message, for example, the first paragraph from a book. Ask learners to split it into several shorter messages to send. Each shorter message needs a maximum of 10 characters, the number of the message (the first part is 1, the second 2, etc.), the destination (the learner to send the message to), and their own name. Ask learners to send their messages to other learners, so that they are passed around the class before getting to their destination.

Main teaching ideas

1 Suitability of data transmission (20 minutes)

Learning intention: Explain the suitability of each method of data transmission for a given scenario.

Resources: Example situations.

Description and purpose: Put learners into pairs and give each pair a different context. Ask the learners to work in pairs to work through each method of data transmission in turn, and justify either why they are, or are not, suitable. Ask each pair to read their context, and then their choices for each method. Encourage other learners to challenge their responses and ask questions.

› **Differentiation ideas:**

Support – put learners with a stronger learner. Give learners a set of questions to consider for each method, for example, does the data need to be sent in both directions?

Challenge – ask learners to justify for and against each method, so they are not only justifying if they are, or are not, suitable – but also considering both sides for each.

2 Types of transmission (20 minutes)

Learning intention: Understand how data is transmitted using different methods of transmission.

Resources: Worksheet 2.1, 'Types of data transmission'.

Description and purpose: Ask learners to answer the questions in Worksheet 2.1 to test their understanding of the types of transmission. Review the answers and ask learners to mark and correct their own worksheets.

› **Differentiation ideas:**

Support – give learners fact cards about each type of transmission to use for each question.

Challenge – complete the challenge section of the worksheet to convert their understanding into diagrammatic form.

› **Assessment ideas:** Self-assessment and correction of answers.

3 USB (15 minutes)

Learning intention: Understand the universal serial bus interface and its use.

Resources: A list of questions about USB that increase in difficulty, for example, starting with what it stands for, and ending with justifying its use.

Description and purpose: Give learners a list of question about the USB interface. Ask learners to research the answers to each question and write a response. Work through all the questions in turn, asking different learners for their answers.

Differentiation ideas:

Support – give learners a list of resources, e.g. websites, textbook pages, to use to find their answers.

Challenge – ask learners to justify their choices and expand their answers.

Plenary ideas

1 Application (10 minutes)

Resources: A description of a context that needs a method of transmission identifying.

Description and purpose: Ask learners to work in pairs to identify whether serial or parallel is more appropriate, and whether simplex, half-duplex or full-duplex is more appropriate. Ask learners to vote for their choice, and select learners to justify their choice.

Reflection ideas: Ask learners to consider how they made their choice, and whether they justified the choice they made, and if not how they could do this next time.

2 Key data transmission cards (10 minutes)

Resources: Set of keywords for terms introduced in this section.

Description and purpose: Ask learners to write definitions on the back of each key term and join them together to create a set of definitions for this section.

Homework ideas

1 Transmission method contexts

Description: Give learners a set of scenarios, for example, data is transferred from a school computer to a server. Ask learners to identify whether it should use serial or parallel transmission. Ask learners to identify whether it should use simplex, half-duplex or full-duplex transmission. Ask learners to justify each decision. Review learners' answers in the next lesson.

2 Packet components

Description: Give learners example data from some packets, and the names of the parts, e.g. the header, payload, packet number, etc. Ask learners to identify these features in the packet contents they have been given.

2.2 Methods of error detection

This section covers Coursebook sub-chapter topic 2.5.

LEARNING PLAN

Syllabus learning objectives	Learning intentions	Success criteria
• Understand the need to check for errors after data transmission and how these errors can occur.	• Explain why error checking is required. • Describe how errors in data transmission can occur.	Learners will be able to describe the need for error checking. Learners will be able to identify different ways that errors can occur.

CONTINUED

Syllabus learning objectives	Learning intentions	Success criteria
• Describe the processes involved in each of the following error detection methods for detecting errors in data after transmission: parity check (odd and even), checksum and echo check.	• Describe how parity check can identify errors. • Describe how checksum can detect errors. • Describe how echo check can detect errors.	Identify that parity, checksum and echo checks can be used to check for errors. Learners will be able to describe how parity check works and how it can identify errors. Learners will be able to describe how checksum works and how it can identify errors. Learners will be able to describe how echo check can detect errors.
• Describe how a check digit is used to detect errors in data entry and identify examples of when a check digit is used, including international standard book numbers (ISBN) and bar codes.	• Describe how a check digit can identify errors. • Identify examples of the use of check digits.	Learners will be able to identify checksum as a method of detecting errors in data entry. Learners will be able to describe the processes within check digit. Learners will be able to identify examples where check digits are used.
• Describe how an automatic repeat request (ARQ) can be used to establish that data is received without error.	• Describe the purpose and function of an ARQ.	Learners will be able to define ARQ. Learners will be able to describe how ARQ works.

Common misconceptions

Misconception	How to elicit	How to overcome
Parity, checksum and echo check can identify any and all errors that may occur.	Give learners an example of each where they might state there is no error, but when there is.	Show learners the process and why it does not detect the error.
Check digit is used in data transmission.	Ask learners to identify error detection methods for transmission, and for data entry.	Explain that a check digit is not used in transmission, only data entry.
Check digit and checksum are the same.	Ask learners to describe each method and when they are used.	Explain that check digit is used in data entry, while checksum is used in transmission.
Automatic repeat request is the same as echo.	Ask learners for descriptions of each method.	Explain that ARQ does not send the data back to the transmitting device, it sends a signal to say the data is correct or incorrect.

Starter ideas

1 Error checking (10 minutes)

Resources: A series of bytes of data before sending, and after sending (where several of these have changed).

Description and purpose: Ask learners to work in pairs to come up with a way that could detect the errors in the data transmission. Ask learners to describe their methods, and write these on the board. If any of these are similar to those needed (most likely echo check) then explain this to learners.

What to do next: Error-checking methods, starting with the one(s) most similar to those identified by learners.

2 How can errors occur? (10 minutes)

Description and purpose: Ask learners to work in pairs to generate a list of ways that an error can occur during data transmission. Ask each group for their answers and create a class list.

What to do next: Parity check.

Main teaching ideas

1 Parity (20 minutes)

Learning intention: Understand the processes involved in parity check.

Resources: Worksheet 2.2, 'Parity'.

Description and purpose: Ask learners to complete the questions on the worksheet. When finished ask learners to compare their answers with another learner and discuss any differences. Ask learners for the answers to each question and review these as a class.

› **Differentiation ideas:**

Challenge – ask learners to work out if there is a possible solution to the situation identified in the challenge section.

› **Assessment ideas:** Peer assess each other's answers, then self assess against the correct answers.

2 Check digit (20 minutes)

Learning intention: Understand how a check digit is used to detect errors in data entry.

Resources: Example algorithms for calculating a check digit.

Description and purpose: Show learners how the first algorithm is used to calculate a check digit. Ask learners to perform the same algorithm on their set of data. Ask learners to compare their check digits with others and identify if there are any errors. Repeat this with other common check digit algorithms. Ask learners to act as the sender and receiver. Learner 1 calculates the check digit with their algorithm, then sends the data and check digit to learner 2. Learner 2 recalculates the check digit to identify if there are any errors.

› **Differentiation ideas:**

Support – give learners a structure worksheet that takes them through the calculation step-by-step with spaces to write their results.

3 Automatic repeat request (10 minutes)

Learning intention: Understand the processes involved in automatic repeat request.

Resources: Set of instructions followed by the automatic repeat request cut into individual statements.

Description and purpose: Give each pair the set of instructions that are separated. Ask learners to put them back into the correct order. Ask the pairs for the next one in the sequence and display the correct order on the board. Give learners a list of the statements in the correct order to memorise.

› **Differentiation ideas:**

Support – put learners with a strong learner.

Challenge – ask learners to consider whether there is more than one correct order for the statements, or whether this would then perform a different function.

Plenary ideas

1 Parity (10 minutes)

Resources: Sets of data that have been sent and received without error using a specific parity (e.g. odd or even).

Description and purpose: Ask learners to identify the correct parity bit for each set of data. Review each answer by asking learners to vote whether it should be a 1 or a 0.

2 Uses of check digit (10 minutes)

Description and purpose: Ask learners to find as many different examples where a check digit is used in 5 minutes, for example, by searching the internet. Ask learners for their points and to add up the total number of valid examples.

Homework ideas

1 Error detection mind map

Description: Ask learners to create a mind map about error detection. Error detection should be at the start and then the related sections coming from this.

2 Explaining to younger learners

Description: Tell learners that a group of 10-year-olds need to understand how errors can occur in data transmission and entry, as well as methods of detecting these errors. Learners need to write a 10-year-old friendly report that includes reasons why errors can occur, and how these can be identified.

2.3 Encryption

This section covers Coursebook sub-chapter topic 2.6.

LEARNING PLAN

Syllabus learning objectives	Learning intentions	Success criteria
• Understand the need for and purpose of encryption when transmitting data.	• Explain why encryption is used when transmitting data.	Learners will be able to explain the need for, and purpose of, encryption.
• Understand how data is encrypted using symmetric and asymmetric encryption.	• Explain how symmetric and asymmetric encryption work.	Learners will be able to describe symmetric and asymmetric encryption.

Common misconceptions

Misconception	How to elicit	How to overcome
Learners may think that an encrypted file cannot be read. The encrypted data can be read, but it will not make any sense.	Ask learners to define encryption and its purpose.	Explain that the encrypted data will be composed of character and/or symbols that can be seen and read. Encryption means the original data cannot be read, or understood.
Encrypted data is secure and cannot be decrypted without a key.	Give learners a message that is encrypted, but that they can decrypt without needing a key (e.g. moving each letter along 1-place).	Explain that just because data is encrypted does not mean it is completely secure and that it could still be accessed.

Starter ideas

1 What is encryption? (10 minutes)

Resources: An encrypted message, for example, a word.

Description and purpose: Ask learners to work in pairs to try and decrypt the message. Ask learners how they tackled the problem, and if anyone did decrypt it. Give learners a clue if no-one has managed it.

What to do next: Main teaching idea 1, 'Basic encryption'.

2 Definition matching (10 minutes)

Resources: List of key terms relating to encryption, and definitions.

Description and purpose: Ask learners to match the terms with the definitions. Ask learners for the terms for each definition and explain the answers where needed.

Main teaching ideas

1 Basic encryption (15 minutes)

Learning intention: Understand how data is encrypted.

Resources: Worksheet 2.3, 'Encryption'.

Description and purpose: Ask learners to use the worksheet to decrypt the message and create their own encryption key.

› **Differentiation ideas:**

Challenge – tell learners that this method is very simple to decrypt even without the key. Ask learners if they can think of a method that would be more difficult.

2 Spreadsheet or programming challenge (60 minutes)

Learning intention: Understand how data is encrypted.

Resources: Spreadsheet software or programming software.

Description and purpose: Ask learners to create either a spreadsheet, or a computer program, that will read in a message and then encrypt it. Learners can allow the input in any form, e.g. one letter at a time at a lower level, and the whole message at the same time for higher level. Ask learners to test each other's programs.

› **Differentiation ideas:**

Support – give learners a spreadsheet or program that has sections already written for the entry of the data, and comments identifying what the learner needs to do.

Challenge – ask learners to write an efficient program, for example, using loops instead of repeated statements, and implementing subroutines.

› **Assessment ideas:** The produced systems can be assessed for their function and efficiency. Learners could be given a list of requirements that they have to meet, then the system can be compared with this.

3 Asymmetric vs symmetric encryption

Learning intention: Understand how data is encrypted using symmetric asymmetric encryption.

Resources: List of statements about asymmetric and symmetric encryption.

Description and purpose: Ask learners to work in pairs to identify whether each statement applies to asymmetric encryption, symmetric encryption, or both. Review each statement in turn, asking learners to vote which group they put this in.

› **Differentiation ideas:**

Support – put learners with a strong learner, and provide them with some facts about each method of encryption to use to help.

Challenge – ask learners to add any additional points that are relevant to one, or both, methods.

› **Assessment ideas:** Self-assessment when reviewing the answers.

Plenary ideas

1 Encryption fact cards (10 minutes)

Resources: Two blank cards (e.g. 10 × 15 centimetres) for each learner.

Description and purpose: Ask learners to summarise the need for encryption on one card, and the use of asymmetric and symmetric encryption on the second. Ask learners to compare their points with another learner, to identify anything they have missed.

2 Quick test (10 minutes)

Resources: Set of multiple-choice questions about encryption.

Description and purpose: Ask learners a series of questions about encryption with several different possible answers. Ask learners to write down their answer to each question. Review the questions and ask learners to mark their own answers.

Assessment ideas: Self-assessment when marking answers.

Homework ideas

1 Extending encryption

Description: Ask learners to take their spreadsheet or program from Main teaching idea 2 home with them, and add a second method of encryption to it, and that the user can select which method to use.

2 Encryption mind map

Description: Ask learners to create a mind map about encryption, including symmetric and asymmetric encryption.

Name _____ Date _____

1.1 Hexadecimals

Part 1 – binary to hexadecimal

Getting started

1 0110 _____

2 0000 _____

3 1100 _____

4 1010 _____

Practice

5 1011 0111 _____

6 1111 1101 _____

7 0010 0001 0000 _____

8 1011 1111 1010 _____

Challenge

9 1111000010101111 _____

10 1000011101100001 _____

Part 2 – hexadecimal to binary

Getting started

1 5 _____

2 A _____

3 E _____

Practice

4 21 _____

5 10 _____

6 AB _____

7 D5 _____

Challenge

8 1EF _____

9 20B _____

10 ABCD _____

Name _____ Date _____

1.2 Two's complement

Part 1

Convert the two's complement numbers to denary, and the denary numbers to two's complement.

	Positive two's complement	Denary
Getting started	0111	
		6
	01000	
		9
Practice	01110001	
		38
	01000001	
		89
Challenge	01010111	
		100
	00111111	
		127

Part 2

Convert the negative two's complement binary to denary.

	Negative two's complement	Working	Denary
Getting started	1001		
	1010		
	10011		
	101011		
Practice	11100101		
	11001011		
	10001111		
	10101010		
Challenge	11010101		
	10011110		
	10000001		
	10000000		

Part 3

Convert the denary to negative two's complement binary.

	Denary	Working	Negative two's complement
Getting started	−2		
	−10		
	−24		
	−35		
Practice	−58		
	−61		
	−75		
	−89		
Challenge	−100		
	−116		
	−124		
	−130		

Name _____ Date _____

1.3 Crack the code

Getting started

Use the code in the table below to work out the message below.

a	b	c	d	e	f	g	h	i	j
1	2	3	4	5	6	7	8	9	10

k	l	m	n	o	p	q	r	s	t
11	12	13	14	15	16	17	18	19	20

u	v	w	x	y	z	
21	22	23	24	25	26	27

8 5 12 12 15 27 3 15 13 16 21 20 9 14 7 27 9 19 27 6 21 14

Practice

Finished already? Convert each number in the message above into 8 bit binary.

Challenge

Answer the following questions:

1 If each character can have only 4 bits, how many different characters can you have?

2 If each character can have 8 bits, how many different characters can you have?

3 Create a set of codes that use 8 bits for capital letters and numbers. You cannot use the same code twice, and you cannot use any of the numbers used in the code at the start of this worksheet.

Name _____ Date _____

2.1 Types of data transmission

Getting started

Fill in the gaps about the different types of data transmission.

Data can be transmitted as serial or _____ .

Simplex transmission only transmits data in _____ direction.

Full-duplex can transmit data in _____ directions at the _____ time.

Practice

Tick (✓) **one** box in each row to identify if the statements are true or false.

	True	**False**
Serial transmission always sends data 1 bit at a time.		
Full-duplex always sends multiple bits at a time.		
Use simplex when you need to send data in both directions.		
Parallel transmission always sends 2 bits at a time.		
Use half-duplex when data is needs to go in both directions but not at the same time.		
Simplex transmission can only go in one direction.		
Parallel transmission always uses full-duplex.		

Challenge

1 Draw a labelled diagram to describe serial and parallel data transmission.

2 Draw a labelled diagram to describe simplex, half-duplex, and full-duplex data transmission.

Name _____ Date _____

2.2 Parity

Getting started

All of the following bytes have been sent and received without error, using **even** parity.
Complete the parity bit for each byte.

Parity bit	Data
	1 0 1 1 0 1 0 1
	0 0 1 1 1 0 0 0
	1 0 1 0 1 0 1 0
	0 0 0 0 0 0 0 0
	1 1 1 1 1 1 1 1
	0 1 0 1 0 1 0 0

Practice

1 All of the following bytes have been sent correctly. Identify if odd or even parity is used.

Data
1 0 1 1 0 0 0 0 1
0 1 0 1 1 1 1 0 1
0 0 1 1 0 0 0 0 0

2 All of the following bytes have been sent incorrectly. Identify if odd or even parity is used.

Data
1 1 0 1 1 1 0 0 0
0 1 0 0 1 0 0 1 0
1 0 0 0 0 0 0 0 0

3 One of these bytes has been sent incorrectly. Identify if odd or even parity is used.

Data
1 0 0 1 0 0 0 0 0
0 1 0 0 1 1 0 0 0
0 1 1 0 1 1 0 0 0

Challenge

Identify a situation when parity might not detect an error.

Name _____ Date _____

2.3 Encryption

The table shows the original characters of a message on the first line, and the encrypted character on the second.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g

Getting started

Decrypt the following encrypted message;

lujyfwapvu johunlz aol khah zv pa jhuuva il buklyzavvk dpaovba aol rlf

_____

_____

Practice

Create your own encryption key:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z

Challenge

What will happen if you want to include capital letters, numbers and/or symbols in the message? How would you encrypt this data?

Have a go and create your own encryption method.