#

# KS4 – Data representations

## Unit introduction

Data can be represented in many different forms. You can see this happening throughout time, for example, in the use of cave paintings and clay tablets, through to the use of Morse code. Data and instructions in a computer are formed using a series of 1s and 0s. In this unit, learners will discover how numbers, letters, images, and sound are represented with 1s and 0s. They will also learn about the factors that impact on the quality of those representations, such as bit depth. Finally, learners will be introduced to the concept of compression and discover how to perform run length encoding and Huffman coding as forms of lossless compression.

## Schools in England

This unit has been mapped to the GCSE computer science specifications of all major examination boards in England: AQA (A), OCR (O), Edexcel (Ex), and Eduqas (Eq). You can use the checklist next to each lesson below to see the lessons that relate to your examination board’s specification. You can find a detailed breakdown of the learning objectives in the ‘Data representation objectives’ spreadsheet provided with the lesson resources for Lesson 1.

## Overview of lessons

| **Lesson** | **Brief overview** | **Learning objectives** | **A** | **O** | **Ex** | **Eq** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 What is representation? | Learners will discuss examples of representation from storytelling to clay etchings. They will then be introduced to how data is represented in computers. Next, they will examine a cake recipe and decide which elements are data and which are instructions. Finally, they will think about how you could represent a message using only two states: on and off.  | * Give examples of the use of representation
* Explain that computers use binary to represent all data and instructions
* Explain how binary relates to two-state electrical signals
 | ✔ | ✔ | ✔ | ✔ |
| 2 Number bases | Learners will be reminded of how the base-10 number system works, before being presented with the base-2 number system. They will then complete some conversions from binary to decimal and decimal to binary. | * Explain the difference between base-2 and base-10 numbers
* Convert between binary and decimal numbers
 | ✔ | ✔ | ✔ | ✔ |
| 3 Binary addition | Learners will discover how to count in binary so that they can establish an understanding of the patterns that occur when numbers increase by 1 in binary. They will then be shown the four rules of binary addition, before practising adding two binary numbers together. Finally, learners will move on to adding three binary numbers together.  | * Count in binary
* Perform addition in binary on two binary numbers
* Perform addition in binary on three binary numbers
 | ✔ | ✔ | ✔ | ✔ |
| 4 Binary subtraction | Learners will be introduced to the four rules of binary subtraction. They will then perform subtraction on a series of binary numbers. | * Perform subtraction in binary
 |  |  |  | ✔ |
| 5 Binary shifts | Learners will be introduced to binary shifting that can be used for multiplication and division of binary numbers. They will discover why binary shifting might be needed, and they will learn about overflow and underflow.  | * Perform binary shifts
* Describe situations where binary shifts can be used
* Explain how overflow errors can occur
* Explain how underflow occurs
 | ✔ | ✔ | ✔ | ✔ |
| 6 Signed binary integers | Learners will be introduced to the concept of signed and unsigned integers. They will find out how to identify the least significant and most significant bits. They will learn how to identify a positive and negative integer using sign and magnitude. Finally, they will discover two’s complement.  | * Compare signed and unsigned integers
* Use sign and magnitude to represent positive and negative integers
* Use two’s complement to represent positive and negative integers
 |  |  | ✔ | ✔ |
| 7 Hexadecimal | Learners will be reminded about base 2 and base 10, then they will be introduced to hexadecimal, which is base 16. They will be shown the traditional method for converting between base 2 and base 16, then they will learn about a simpler method that uses two tables to simplify the process.  | * Explain why and where hexadecimal notation is used
* Explain how numbers are represented using hexadecimal
* Convert decimal numbers to and from hexadecimal numbers
 | ✔ | ✔ | ✔ | ✔ |
| 8 Representing text | Learners will be introduced to the ASCII character set. They will learn that it traditionally used 7 bits, but now uses 8 bits. They will calculate the maximum number of characters that can be represented using 7 bits, then they will be introduced to the ASCII table. They will then practise coding words using decimal and binary numbers.  | * Determine the maximum number of states that can be represented by a binary pattern of a given length
* Explain how ASCII is used to represent characters, and its limitations
* Explain what a character set is
* Describe how character codes are commonly grouped and run in sequence within encoding tables
 | ✔ | ✔ | ✔ | ✔ |
| 9 Unicode and file size calculation | The lesson begins with revisiting ASCII and its limitations, namely that it can only represent 128 characters. You will then introduce Unicode as an alternative for a more universal character set. Learners will then calculate the number of bits that are needed to store a piece of text using ASCII and Unicode.  | * Explain the need for Unicode
* State that Unicode uses the same codes as ASCII up to 127
* Calculate the number of bytes needed to store a piece of text
 | ✔ | ✔ |  | ✔ |
| 10 Representing bitmap images | First, learners will investigate what a pixel is by looking at a pixelated image. They will then discover how colour depth and resolution are used to determine the number of available colours and the image size. Finally, learners will find out about metadata and the types of metadata used with a bitmap image.  | * Describe what a pixel is and how pixels relate to bitmap images
* Describe colour depth and resolution
* Define ‘metadata’
* Give examples of metadata applied to a bitmap image
 | ✔ | ✔ | ✔ | ✔ |
| 11 Bitmap file size calculation | First, learners will recap their understanding of bitmap images. They will then discover how to calculate the file size of a bitmap image using different examples. Then, they will investigate how the number of pixels and colour depth can affect the file size of an image.  | * Calculate the file size of bitmaps
* Describe how the number of pixels and colour depth can affect the file size of a bitmap image, using examples
 | ✔ | ✔ | ✔ | ✔ |
| 12 Representing sound | First, learners will be reminded of how sound travels. This will then lead into an explanation of how sound can be interpreted digitally. Through examples and demonstrations, learners will find out about the key terms: sampling, sample rate, and sample resolution.  | * Explain why analogue sound data needs to be converted into binary digits
* Describe the concepts of sampling, sample rate, and sample resolution
 | ✔ | ✔ | ✔ | ✔ |
| 13 Sound file size calculation | First, learners will recap their understanding of how sound is interpreted digitally. They will then learn how to calculate the file size of a sound file through demonstrations and examples. Finally, they will discover how sample rate, duration, and bit depth affect the quality of the sound recorded. **Note:** Learners will use Audacity to help them understand some of the key points addressed in this lesson.  | * Calculate file size requirements of sound files
* Describe the effect of sample rate, duration, and sample resolution on the playback quality and the size of a sound file
* Give examples of metadata applied to sound files
 | ✔ | ✔ | ✔ | ✔ |
| 14 Measurements of storage | Learners will already be familiar with the terms ‘bit’, ‘nibble’, and ‘byte’. They will now learn about other types of storage capacity, and they will practise converting between units of measurement. They will also learn about the difference between storage capacities such as ‘megabyte’ and ‘mebibyte’ to help support them when performing calculations.  | * Define the terms ‘bit’, ‘nibble’, ‘kilobyte’, ‘megabyte’, ‘gigabyte’, ‘terabyte’, and ‘petabyte’
* Compare ‘kibibyte’, ‘mebibyte’, ‘gibibyte’, and ‘tebibyte’ to ‘megabyte’, ‘gigabyte’, and ‘terabyte’’
* Convert between units of measurement
 | ✔ | ✔ | ✔ | ✔ |
| 15 Lossy and lossless compression | Learners will be introduced to the concept of data compression and learn about why it is needed, and that it comes in different forms. They will be introduced to the terms ‘lossy’ and ‘lossless’.  | * Explain what data compression is
* Explain why data may be compressed, and that there are different ways to compress data
* Define ‘lossy compression’ and ‘lossless compression’
 | ✔ | ✔ | ✔ | ✔ |
| 16 Run length encoding | Learners will be introduced to run length encoding (RLE), which is a type of lossless compression. Through demonstrations and examples, learners will find out about frequency pairs and manually perform RLE on some data. Finally, they will learn about the term ‘compression ratios’ and calculate the compression ratios for different file sizes. | * Explain how data can be compressed using run length encoding (RLE)
* Represent data in RLE frequency/data pairs
* Calculate compression ratios
 | ✔ |  |  | ✔ |
| 17 Huffman coding | Learners will be introduced to Huffman coding, which is another form of lossless compression. They will discover how the algorithm works and practise applying it to some sample data, and they will practise interpreting Huffman trees.  | * Explain how data can be compressed using Huffman coding
* Interpret a Huffman tree
* Calculate the number of bits required to store a piece of data compressed using Huffman coding
 | ✔ |  |  |  |
| 18 Summative assessment | This is the final lesson of the unit, and is an opportunity for learners to confirm their understanding of the content covered in the unit. The assessment consists mostly of closed-form questions with a few longer, exam-style questions towards the end.  | * Summarise learning through a final assessment
 | ✔ | ✔ | ✔ | ✔ |

## Progression

This unit progresses learners’ knowledge and understanding of data representations and compression. View the learning graphs to see clear progression routes.

## Curriculum links

[**National curriculum links**](https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study)

* develop their capability, creativity and knowledge in computer science, digital media and information technology
* develop and apply their analytic, problem-solving, design, and computational thinking skills

## Assessment

### Summative assessment

* This unit includes a final summative assessment to be used at the end of the unit. An answer sheet is also provided.

## Subject knowledge

This unit focuses on data representations and compression.

Enhance your subject knowledge to teach this unit through the following training opportunities:

### Online training courses

* [Data Representation in Computing: Bring Data to Life](https://www.futurelearn.com/courses/representing-data-with-images-and-sound)
* [How Computers Work: Demystifying Computation](https://www.futurelearn.com/courses/how-computers-work)

### Face-to-face and remote courses

* [An introduction to algorithms, programming and data in computer science (face-to-face)](https://teachcomputing.org/courses/CP228/an-introduction-to-algorithms-programming-and-data-in-gcse-computer-science-face-to-face)
* [An introduction to algorithms, programming and data in computer science (remote)](https://teachcomputing.org/courses/CP428/an-introduction-to-algorithms-programming-and-data-in-computer-science-remote)
* [Maths in computer science (face-to-face)](https://teachcomputing.org/courses/CP234/maths-in-computer-science-face-to-face)
* [Maths in computer science (remote)](https://teachcomputing.org/courses/CP434/maths-in-computer-science-remote)

Resources are updated regularly — the latest version is available at: [ncce.io/tcc](http://ncce.io/tcc).

This resource is licensed under the Open Government Licence, version 3. For more information on this licence, see [ncce.io/ogl](https://ncce.io/ogl).