Grade 11 Computer Studies

CBC NOTES | Terms 1, 2 and 3 - notes for 11 topics

IN EVERY TOPIC

- What you will learn, the key words and key terms explained.
- Clear notes under headings and subheadings, with labelled pictures and step-by-step procedures.
- Comparison tables, and advantages and disadvantages where methods are compared.
- Key points to remember, then structured questions to practise.
- All the answers and a glossary at the back.

TOPICS IN THESE NOTES

Topic	Name	Term
1.1	Data Representation	1
1.2	Binary Arithmetic Operations	1
1.3	Logic Gates	1
1.4	Logic Circuits	1
2.1	Computer Network Standards and Protocols	1
2.2	Network Security	2
3.1	Object Oriented Programming (OOP) Concepts	2
3.2	Object Oriented Programming Implementation	2
3.3	Relational Database Fundamentals	3
3.4	Structured Query Language (SQL)	3
3.5	Web Development Fundamentals	3

A TASTE OF THE FIRST TOPIC

Data Representation

A photo, a song, a text message and a video all sit side by side on your phone, yet inside the phone they are nothing but 0s and 1s. In this topic we learn the units used to measure data, how text, images, sound and video are stored, the decimal, binary, octal and hexadecimal number systems and where each is used, how to convert between them, the coding schemes ASCII, EBCDIC and BCD, and why good data representation matters.

CONTENTS

Topic	Name	Term	Page
1.1	Data Representation	1	3
1.2	Binary Arithmetic Operations	1	9
1.3	Logic Gates	1	13
1.4	Logic Circuits	1	16
2.1	Computer Network Standards and Protocols	1	20
2.2	Network Security	2	25
3.1	Object Oriented Programming (OOP) Concepts	2	30
3.2	Object Oriented Programming Implementation	2	34
3.3	Relational Database Fundamentals	3	41
3.4	Structured Query Language (SQL)	3	45
3.5	Web Development Fundamentals	3	50
	Answers		57
	Glossary		61

Elimu254
Learn • Grow • Build Tomorrow

Strand 1.0 Foundation of Computer Studies | Term 1

1.1 Data Representation

WHAT YOU WILL LEARN

By the end of this topic you should be able to:

- identify the units used to represent data in computers
- describe the representation of different types of data in computers
- describe the types of number systems used in computers
- relate different number systems to their real-world applications in computing
- convert numbers to different number systems
- represent data using different coding schemes
- recognise the importance of data representation in computers

KEY WORDS

bit, nibble, byte, word, pixel, sampling, number system, radix, positional value, decimal, binary, octal, hexadecimal, coding scheme, ASCII, EBCDIC, BCD.

A photo, a song, a text message and a video all sit side by side on your phone, yet inside the phone they are nothing but 0s and 1s. In this topic we learn the units used to measure data, how text, images, sound and video are stored, the decimal, binary, octal and hexadecimal number systems and where each is used, how to convert between them, the coding schemes ASCII, EBCDIC and BCD, and why good data representation matters.

1. Units of data

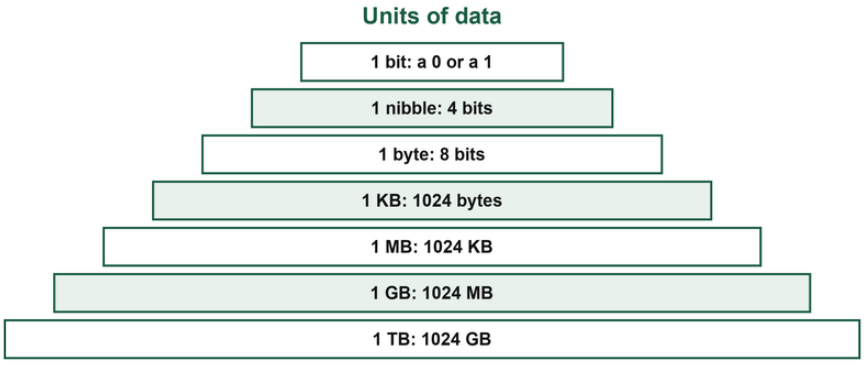
A computer is built from millions of tiny switches (transistors), each either on or off. So a computer works with two symbols: 1 for on and 0 for off.

Bit
A binary digit: the smallest unit of data, a single 0 or 1.

Nibble
A group of 4 bits.

Byte
A group of 8 bits, enough to store one character.

Word
The number of bits a processor handles at one time, usually 32 or 64 bits.



Units of data from the bit to the terabyte

Working with units

2 MB in KB: 2 × 1024 = 2048 KB.

64 bits in bytes: 64 ÷ 8 = 8 bytes.

Number of 5 MB songs on a 4 GB flash disk: 4 × 1024 = 4096 MB; 4096 ÷ 5 gives 819 songs.

WATCH OUT

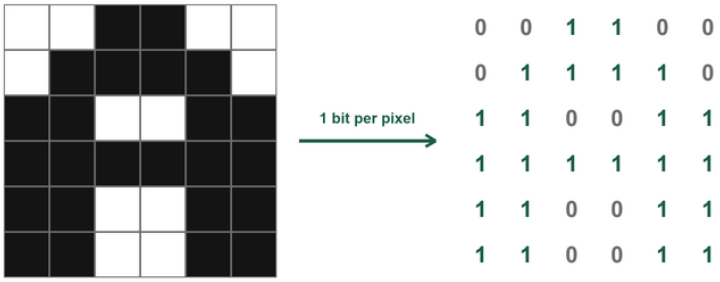
A small b means bits and a capital B means bytes. A speed of 8 Mb/s is 1 MB/s. A 64-bit processor handles more data in one step than a 32-bit one.

2. How different types of data are represented

Representing different types of data

Type of data	How it is stored	What affects the size
Text	Each character is given a number by a coding scheme, stored in binary	Number of characters
Images	A grid of pixels; each pixel stores a colour as a number	Resolution and colour depth (bits per pixel)
Audio	The sound wave is measured many times a second (sampling) and each value is stored	Sampling rate, bit depth and length
Video	Many images (frames) shown quickly, usually 25 or 30 a second, plus sound	Resolution, frame rate and length; usually compressed

A 6 x 6 black and white image stored as bits



A small black and white image stored as bits, one bit per pixel

Size of an image and a sound

An 800 × 600 image at 24 bits per pixel: 800 × 600 × 24 = 11520000 bits; 11520000 ÷ 8 = 1440000 bytes, about 1.4 MB.

Ten seconds of mono sound at 44100 samples a second and 16 bits: 44100 × 16 × 10 = 7056000 bits; 7056000 ÷ 8 = 882000 bytes, about 0.84 MB.

COMPRESSION

JPG, MP3 and MP4 files are small because they remove detail that is hard to notice. This saves storage and speeds up sending files.

3. Number systems

Radix (base)
The number of different digits a number system uses.

Positional value
The value of a digit depends on its position; each place is a power of the radix.

The four number systems

Decimal (base 10)	Binary (base 2)
Digits 0-9. Used by people for prices and marks. 13	Digits 0 and 1. Used inside the computer. 1101

Octal (base 8)	Hexadecimal (base 16)
Digits 0-7. Used for Linux file permissions. 15	Digits 0-9, A-F. Colours, MAC and memory addresses. D

The four number systems, each showing the same value

Equivalent values in the four systems

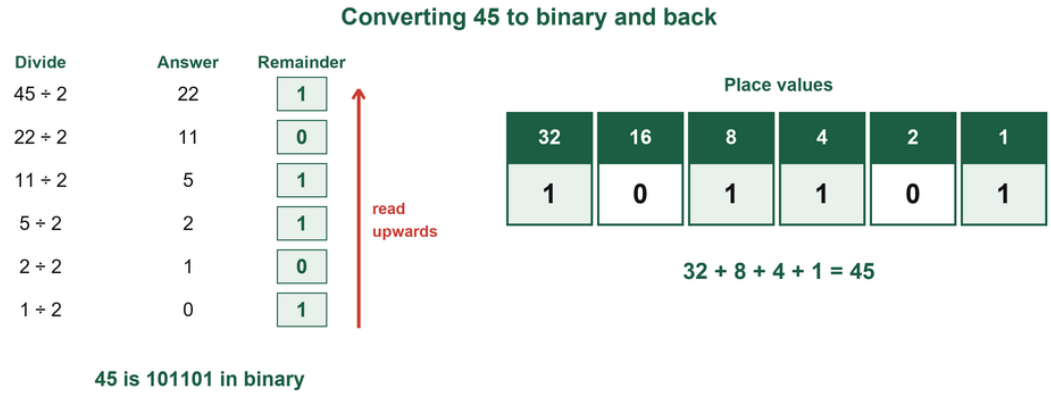
Decimal	Binary	Octal	Hexadecimal
0	0000	0	0
5	0101	5	5
8	1000	10	8
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Where each number system is used

System	Real-world applications
Binary	Inside the computer: machine code, memory, logic circuits
Decimal	What people type and read: prices, marks, ages
Octal	File permissions in Linux, such as 755
Hexadecimal	Web colours (#FF0000 is red), MAC addresses, memory addresses, IPv6, error codes

REMEMBER
Hexadecimal is for people. It is a short way of writing binary: one hexadecimal digit stands for 4 bits. The computer itself stores only binary.

4. Converting between number systems



Repeated division and place values for the number 45

Decimal to any base

1. Divide the number by the base and write the remainder.
2. Divide the answer again, and repeat until the answer is 0.
3. Read the remainders from the BOTTOM up. In hexadecimal, change 10 to 15 into A to F.

Decimal to octal and hexadecimal

156 to octal: 156 ÷ 8 = 19 R 4; 19 ÷ 8 = 2 R 3; 2 ÷ 8 = 0 R 2. Answer: 234(8).
156 to hexadecimal: 156 ÷ 16 = 9 R 12 (C); 9 ÷ 16 = 0 R 9. Answer: 9C(16).

Any base to decimal: add the place values

11011(2): 16 + 8 + 2 + 1 = 27.
157(8): 1 × 64 + 5 × 8 + 7 × 1 = 111.
3A7(16): 3 × 256 + 10 × 16 + 7 × 1 = 935.

Short cuts through binary

Binary to octal: group in threes from the right. 10011100 is 010 011 100, which is 234(8).
Binary to hexadecimal: group in fours from the right. 10011100 is 1001 1100, which is 9C(16).
Octal to hexadecimal: go through binary. 157(8) is 001 101 111; regroup as 0110 1111, which is 6F(16).

A Python program to convert a number

```
def to_base(n, base):
    digits = "0123456789ABCDEF"
    if n == 0:
        return "0"
    result = ""
    while n > 0:
        result = digits[n % base] + result
        n = n // base
    return result


print(to_base(156, 2))
print(to_base(156, 8))
print(to_base(156, 16))
```

The program prints 10011100, then 234, then 9C. n % base is the remainder and n // base is the whole-number answer, the same two results as in the division ladder. Python also has bin(), oct(), hex() and int("9C", 16).

5. Coding schemes

Coding scheme
An agreed table that gives each character a number, so that different computers read data in the same way.

Coding schemes

Scheme	Size	Features	Example
ASCII	7 bits (128 characters); extended ASCII 8 bits (256)	Used on most computers	A is 65 (01000001); a is 97; 0 is 48; a space is 32
EBCDIC	8 bits	Made by IBM for mainframe computers	A is C1(16), which is 11000001; 0 is F0(16)
BCD	4 bits for each decimal digit	Numbers only; used in calculators, clocks and money systems	59 is 0101 1001

PARITY
A parity bit is one simple check on transmission. In even parity the sender adds a bit so that the number of 1s is even. If the receiver counts an odd number, an error has happened. It cannot detect two flipped bits.

Unicode extends these ideas, with enough codes for all the world's languages and symbols.

6. Why data representation matters

- Performance: small, efficient data is processed and sent faster.
- Storage: compression and suitable data types save disk space.
- Security: encryption works on the binary form of data, and exact data formats help to detect tampering.
- Accuracy: a wrong format brings errors. Too few bits for a number can give a wrong result; two systems using different coding schemes will show garbled text.

IN THE COMMUNITY
Explain to people that a phone bundle is measured in MB because everything sent is counted in bytes; a video uses far more data than a text message, because it holds so many more bits.

KEY POINTS TO REMEMBER
- A bit is a 0 or 1; a nibble is 4 bits; a byte is 8 bits; a word is what a processor handles at one time.
- Text uses coding schemes; images use pixels; sound uses sampling; video uses frames.
- Decimal has radix 10, binary 2, octal 8, hexadecimal 16.
- To convert from decimal, divide repeatedly and read the remainders upwards; to convert to decimal, add the place values.
- Group bits in threes for octal and fours for hexadecimal.
- ASCII, EBCDIC and BCD are coding schemes; BCD codes one decimal digit at a time in 4 bits.
- Efficient data representation improves performance, storage, security and accuracy.

ANSWER THESE

1. Distinguish between a bit and a byte. (2 marks)
2. A learner has a 16 GB flash disk. How many 8 MB files can it hold? (2 marks)
3. Calculate the size in bytes of a 100 × 100 image with 8 bits per pixel. (2 marks)
4. Convert 45 to (a) binary (b) octal (c) hexadecimal. (3 marks)
5. Convert 1101110(2) to (a) decimal (b) hexadecimal. (3 marks)
6. Convert 2F(16) to octal, showing your working. (3 marks)
7. Write 906 in BCD and state why BCD is used in digital clocks. (3 marks)
8. Give one use each of binary, octal and hexadecimal numbers in computing. (3 marks)

Elimu254
Learn • Grow • Build Tomorrow