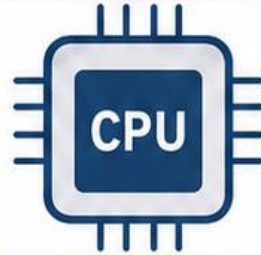


THE CPU: COMPONENTS AND PERFORMANCE

WHAT IS THE CPU?

The Central Processing Unit processes data and instructions by repeatedly carrying out the fetch-decode-execute cycle.



1

FETCH –

Get the next instruction from memory

2

DECODE –

The CU works out what the instruction means

3

EXECUTE –

The instruction is carried out

FOUR COMMON CPU COMPONENTS



ALU –

Arithmetic Logic Unit –

Carries out arithmetic calculations and logical operations.



CU –

Control Unit –

Coordinates the CPU and sends control signals to manage the FDE cycle.



CACHE –

Fast, small memory –

Stores frequently or recently used data and instructions close to the CPU.



REGISTERS –

Tiny, high-speed storage –

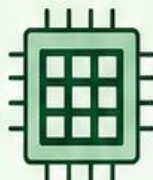
Temporarily hold data, instructions or addresses currently being used.

THREE CPU PERFORMANCE FACTORS



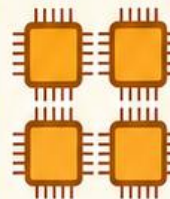
CLOCK SPEED –

Measured in hertz (Hz). A higher clock speed normally completes more FDE cycles each second.



CACHE SIZE –

A larger cache stores more frequently used data and instructions, reducing slower access to RAM.



NUMBER OF CORES –

More cores can process more instructions at the same time when software supports parallel processing.



EXAM TIP – Performance depends on the combination of factors. More cores do not always guarantee a faster program.



ALU calculates • CU controls • cache keeps nearby • registers hold now

VON NEUMANN ARCHITECTURE

The stored-program concept

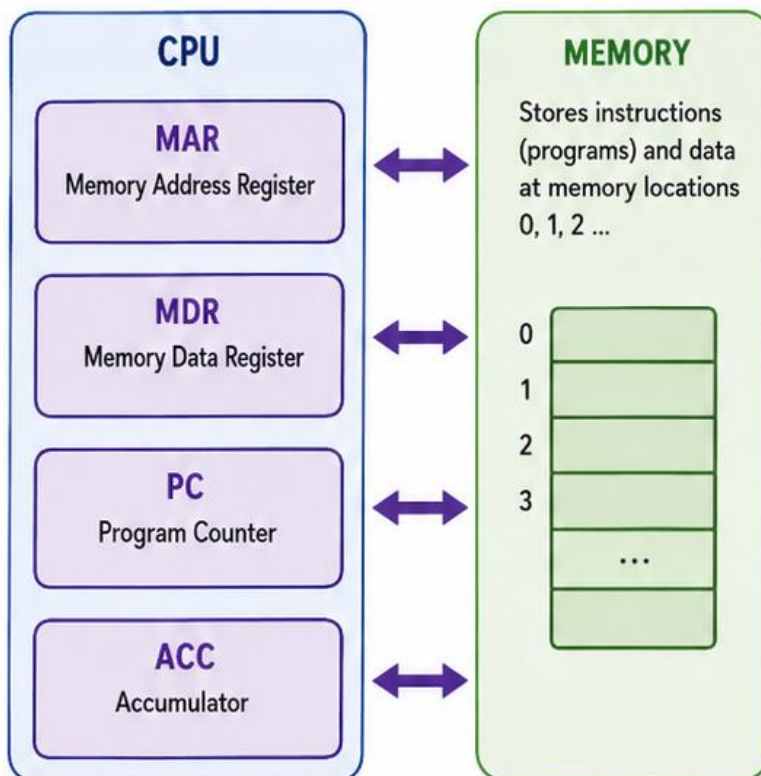


The Von Neumann architecture is a design model for computers. It uses one memory to store both instructions (programs) and data. The CPU and memory communicate using address, data and control signals.

✓ KEY IDEA

Instructions and data are stored in the same memory and are fetched sequentially by the CPU.

THE VON NEUMANN MODEL



THE REGISTERS

MAR – Memory Address Register

Holds the address of the memory location to be accessed.

MDR – Memory Data Register

Holds the data being transferred to or from memory.

PC – Program Counter

Holds the address of the next instruction to be fetched.

ACC – Accumulator

Stores data used in calculations and the results of operations.



KEY POINT

These four registers work together to fetch instructions from memory, process them, and store the results.



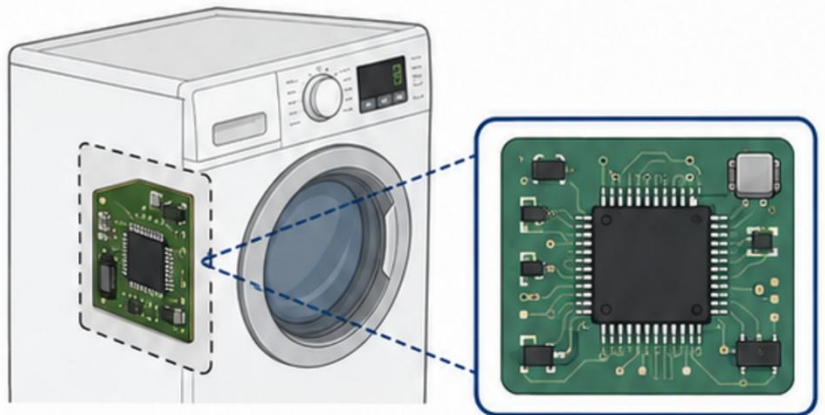
EXAM TIP

You may be asked to identify the purpose of each register and how they work in the fetch-decode-execute cycle.

EMBEDDED SYSTEMS

WHAT IS AN EMBEDDED SYSTEM?

An embedded system is a computer system built into a larger device to perform a specific function.



COMMON CHARACTERISTICS



1. DEDICATED FUNCTION —
Designed to perform one main job.



2. BUILT IN —
Forms part of a larger device or system.



3. LIMITED RESOURCES —
Usually has only the processing power and memory it needs.



4. RELIABLE —
Designed to operate consistently for its intended purpose.



5. REAL-TIME RESPONSE —
Some systems must respond immediately to an input.



6. LIMITED INTERFACE —
May have only buttons, sensors, lights or a small display.

WHERE ARE THEY USED?



Washing machine — controls the wash cycle



Traffic lights — controls the light sequence



Car braking system — monitors and controls braking



Thermostat — controls temperature



Microwave — controls cooking time and power



Fitness tracker — monitors movement and heart rate



EMBEDDED SYSTEM —
one specific function, built into a device

VS

GENERAL-PURPOSE COMPUTER —
performs many different tasks, runs different applications



EXAM TIP

Describe the function performed by the embedded system, not simply the device containing it.

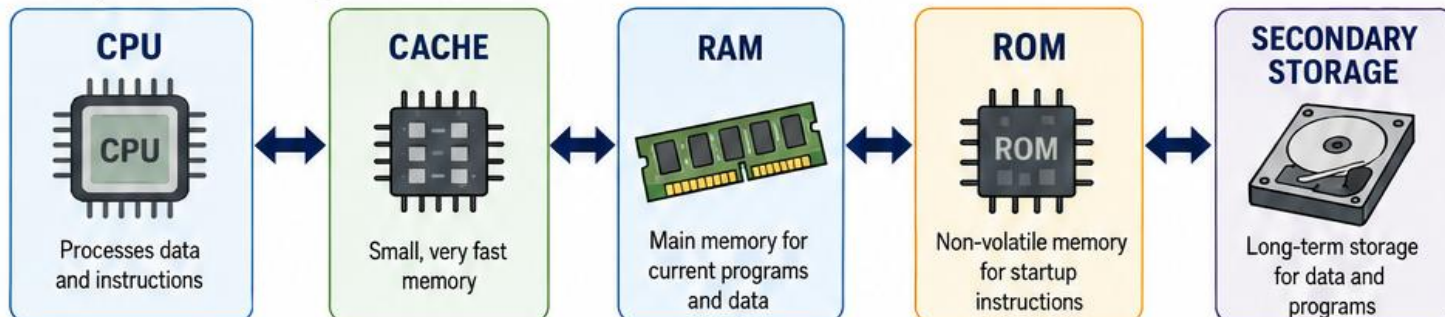


Embedded = built in for one main job

PRIMARY STORAGE

WHY IS PRIMARY STORAGE NEEDED?

Primary storage holds the data and instructions that a computer needs to access directly while it is operating.



RAM AND ROM

RAM — Random Access Memory

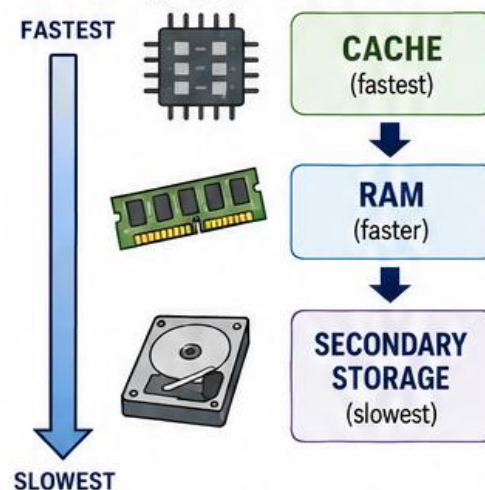
- Volatile — contents are lost when power is removed.
- Stores programs and data currently being used.
- Can be read from and written to.
- More RAM allows more programs and data to be held at once.

ROM — Read Only Memory

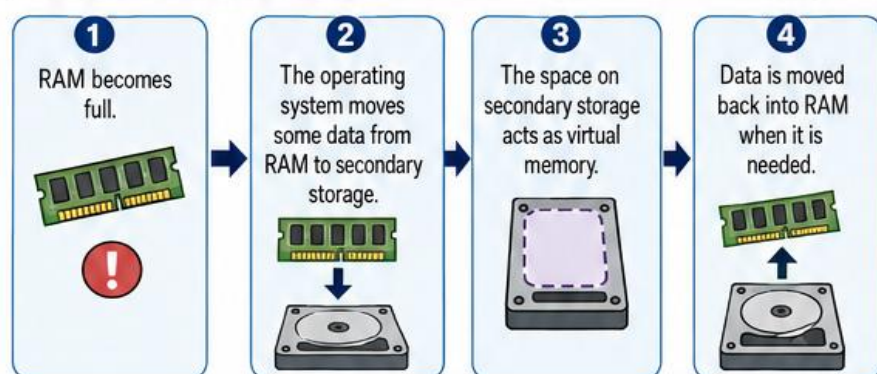
- Non-volatile — contents remain when power is removed.
- Stores instructions needed when the computer starts.
- Contents are normally read rather than changed.

CACHE MEMORY

Cache is small, fast memory located close to or inside the CPU. It stores frequently or recently used data and instructions so the CPU can access them more quickly than from RAM.



VIRTUAL MEMORY



Virtual memory is slower than RAM and may reduce system performance.

CACHE — fastest, smallest 	RAM — fast, temporary working memory 	VIRTUAL MEMORY — slower emergency space on secondary storage 	SECONDARY STORAGE — non-volatile, larger capacity
---	--	--	---



EXAM TIP

Virtual memory does not increase the amount of physical RAM. It uses part of secondary storage when RAM is full.



RAM works now • ROM starts up • cache keeps close • virtual memory helps when RAM fills

SECONDARY STORAGE

Long-term storage for data and programs

WHY DO COMPUTERS NEED SECONDARY STORAGE?



- ✓ Primary memory (RAM) is volatile – data is lost when power is turned off.
- ✓ Secondary storage is non-volatile – it keeps data even without power.
- ✓ Stores large amounts of data and programs for future use.

TYPES OF SECONDARY STORAGE

OPTICAL



Data is read and written using a laser on a disc.

MAGNETIC



Data is stored using magnetic fields on platters or tape.

SOLID STATE



Data is stored on electronic chips with no moving parts.

COMPARING STORAGE DEVICES

DEVICE (EXAMPLES)	CAPACITY	SPEED	PORTABILITY	DURABILITY	RELIABILITY	COST	ADVANTAGES	DISADVANTAGES
OPTICAL (CD, DVD, Blu-ray) 	☆☆☆ Low	☆☆☆ Slow	☆☆☆ High	☆☆☆ Low (scratches easily)	☆☆☆ Medium	☆☆☆ Low	- Cheap - Portable - Good for backups - Read-only discs can't be changed	- Low capacity - Slow - Can be scratched or damaged
MAGNETIC (HDD, Magnetic Tape) 	☆☆☆ High	☆☆☆☆ Medium	☆☆☆ Low (HDD medium)	☆☆☆ Medium (shock sensitive)	☆☆☆ Medium	☆☆☆ Medium	- High capacity - Cost per GB low - Good for large storage needs	- Moving parts can wear out - Slower than SSDs - Less portable (especially HDDs)
SOLID STATE (SSD, USB Flash Drive, Memory Card) 	☆☆☆ High (USB lower)	☆☆☆☆ Fast	☆☆☆ High	☆☆☆ High (no moving parts)	☆☆☆ High	☆☆☆ High	- Very fast - Reliable - Durable - Small and light	- More expensive per GB - Limited write cycles (SSD)

CHOOSING THE RIGHT STORAGE



Consider the application – what is the data being used for?



Balance the trade-offs – capacity, speed, portability, durability, reliability and cost.



Different devices are suited to different tasks and environments.



There is no single "best" storage – it depends on the needs.

APPLY YOUR KNOWLEDGE

Choose the most suitable storage device/media for each scenario and explain why.

- 1 A photographer needs to store very large image files and edit them quickly.
- 2 A school wants to back up important data cheaply and store it off-site.
- 3 A student needs to carry documents between school and home.
- 4 A music producer needs fast access to large sample libraries during recording.
- 5 A company needs to archive data that won't change and must be kept for many years.



EXAM TIP

Always justify your choice by referring to the characteristics: capacity, speed, portability, durability, reliability and cost.

UNITS OF DATA STORAGE

From bits to petabytes – and how to calculate it all

DATA UNITS

	Bit	1 bit
	Nibble	4 bits
	Byte	8 bits
	Kilobyte (KB)	1,000 bytes
	Megabyte (MB)	1,000 KB
	Gigabyte (GB)	1,000 MB
	Terabyte (TB)	1,000 GB
	Petabyte (PB)	1,000 TB



WHY BINARY?



Computers can only understand two states – ON (1) or OFF (0).



All data (text, images, sound, video) must be converted into binary format before it can be processed by a computer.

BINARY CONVERSION EXAMPLES



Text

Each character is represented by 8 bits (1 byte).

A = 01000001



Image

Each pixel is represented by a number of bits (e.g. 24 bits).



Sound

Sampled and stored as a series of binary numbers.

STORAGE DEVICES AND MEDIA

Device / Media	Typical Capacity	Portable	Uses
 USB stick	16 GB – 1 TB	✓	File transfer, backup
 External HDD	1 TB – 8 TB	✓	Large storage, backups
 SSD	256 GB – 4 TB	✓	Fast storage, OS
 SD card	4 GB – 1 TB	✓	Cameras, phones
 Cloud storage	Varies	✓	Online backup, access anywhere

CALCULATION EXAMPLES



Example 1 – Sound File Size

Sample rate = 44,100 Hz
Duration = 120 seconds
Bit depth = 16 bits

File size = $44,100 \times 120 \times 16$
= 84,672,000 bits
= 10,584,000 bytes
≈ 10.1 MB



Example 2 – Image File Size

Colour depth = 24 bits
Width = 3000 px
Height = 2000 px

File size = $24 \times 3000 \times 2000$
= 144,000,000 bits
= 18,000,000 bytes
≈ 17.2 MB



Example 3 – Text File Size

Bits per character = 8
Characters = 2,500

File size = $8 \times 2,500$
= 20,000 bits
= 2,500 bytes



KEY POINTS

- ✓ Computers work with binary data (0s and 1s).
- ✓ Data is stored in units, from bits to petabytes.
- ✓ Storage devices have fixed capacities.
- ✓ You can calculate file sizes and storage needs.



QUICK CHECK

Why must data be stored in binary format?

How many bytes are there in 3 KB?

- What is the size in MB of a 5,000 × 4,000 pixel image at 24 bits?
- You have a 128 GB SSD. Will it be large enough for 10 films, each 4.5 GB? Show your working.

CALCULATING FILE SIZES

THREE FORMULAS TO REMEMBER



SOUND FILE SIZE (bits)

sample rate $\times$ duration in seconds $\times$ bit depth



IMAGE FILE SIZE (bits)

colour depth $\times$ image height in pixels $\times$ image width in pixels



TEXT FILE SIZE (bits)

bits per character $\times$ number of characters

WORKED EXAMPLE 1 – SOUND

Sample rate = 8,000 Hz
Duration = 10 seconds
Bit depth = 8 bits



$$8,000 \times 10 \times 8 = 640,000 \text{ bits}$$

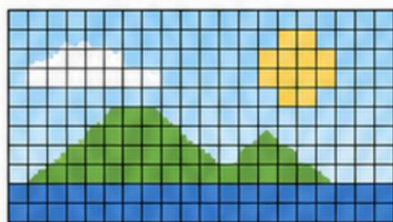
$$640,000 \div 8 = 80,000 \text{ bytes}$$

$$80,000 \div 1,000 = 80 \text{ KB}$$

Answer = 80 KB

WORKED EXAMPLE 2 – IMAGE

Width = 200 pixels
Height = 100 pixels
Colour depth = 24 bits



$$24 \times 100 \times 200 = 480,000 \text{ bits}$$

$$480,000 \div 8 = 60,000 \text{ bytes}$$

$$60,000 \div 1,000 = 60 \text{ KB}$$

Answer = 60 KB

WORKED EXAMPLE 3 – TEXT

Characters = 1,000
Bits per character = 8



Character	8-bit code
A	01000001
B	01000010
C	01000011
...	...

$$8 \times 1,000 = 8,000 \text{ bits}$$

$$8,000 \div 8 = 1,000 \text{ bytes}$$

$$1,000 \text{ bytes} = 1 \text{ KB}$$

Answer = 1 KB

A SAFE CALCULATION METHOD

- 1 Write down the correct formula.
- 2 Substitute the values and calculate the answer in bits.
- 3 Divide by 8 to convert bits into bytes.
- 4 Convert into KB or MB if required and include the unit.

OCR accepts conversions using 1,000 or 1,024.



EXAM TIP

Calculate in bits first. Divide by 8 to convert bits into bytes, then convert into the unit requested.



Formula • substitute • calculate bits • divide by 8 • convert the unit

BINARY & DENARY CONVERSIONS

Convert positive denary whole numbers to binary (up to and including 8 bits) and vice versa.

BINARY NUMBER RANGE: 00000000 – 11111111 • DENARY RANGE: 0 – 255

1. DENARY TO BINARY

Convert a positive denary whole number to binary using division by 2.

- 1 Divide the denary number by 2. Write down the remainder (0 or 1).
- 2 Divide the quotient by 2.
- 3 Repeat until the quotient is 0.
- 4 Read the remainders from bottom to top to get the binary number.

EXAMPLE: 37₁₀ to binary (8 bits)

Divide by 2	Quotient	Remainder
37 ÷ 2	18	1
18 ÷ 2	9	0
9 ÷ 2	4	1
4 ÷ 2	2	0
2 ÷ 2	1	0
1 ÷ 2	0	1

Read
bottom
to top

Remainders (bottom to top): 1 0 1 0 0 1

37₁₀ = 0010 0101₂ (8 bits)

2. BINARY TO DENARY

Convert a binary number (up to 8 bits) to denary by multiplying each bit by its place value (power of 2) and adding the results.

PLACE VALUE TABLE (8 BITS)

Place value (2 ⁿ)	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
Value	128	64	32	16	8	4	2	1

EXAMPLE: 1010 1101₂ to denary

Place value	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
Value	128	64	32	16	8	4	2	1
Binary digits	1	0	1	0	1	1	0	1
Calculation	128 (128×1)	0 (64×0)	32 (32×1)	0 (16×0)	8 (8×1)	4 (4×1)	0 (2×0)	1 (1×1)

Add the values: 128 + 32 + 8 + 4 + 1 = 173

So, 1010 1101₂ = 173₁₀

QUICK REFERENCE: DENARY ⇌ BINARY (8 BITS)

Denary	Binary (8 bits)	Denary	Binary (8 bits)	Denary	Binary (8 bits)
0	00000000	8	00001000	10	00001010
1	00000001	16	00010000	50	00110010
2	00000010	32	00100000	100	01100100
3	00000011	64	01000000	150	10010110
4	00000100	128	10000000	180	10110100
5	00000101	200	11001000	210	11010010
6	00000110	240	11110000	250	11111010
7	00000111	255	11111111		

KEY POINTS

- ✓ Binary uses only two digits: 0 and 1.
- ✓ Each position (bit) represents a power of 2.
- ✓ The leftmost bit is 2⁷ (128). The rightmost bit is 2⁰ (1).
- ✓ Always write binary answers using 8 bits (add leading zeros if needed).
- ✓ Denary range 0–255 fits in 8 bits.

PRACTICE YOURSELF

A. CONVERT DENARY TO BINARY (8 BITS)

Convert the following denary numbers to binary.

- 1 5₁₀ = _____₂
- 2 23₁₀ = _____₂
- 3 128₁₀ = _____₂
- 4 201₁₀ = _____₂
- 5 255₁₀ = _____₂

B. CONVERT BINARY TO DENARY

Convert the following binary numbers to denary.

- 1 00011101₂ = _____₁₀
- 2 01010110₂ = _____₁₀
- 3 10000000₂ = _____₁₀
- 4 11110000₂ = _____₁₀
- 5 10110101₂ = _____₁₀

REMEMBER

Binary number range:
00000000 – 11111111 (8 bits)



Denary number range:
0 – 255



Memory Tip:

Powers of 2 double as we move left.

BINARY & HEXADECIMAL CONVERSIONS

Convert binary integers to their hexadecimal equivalents and vice versa.

BINARY RANGE: 00000000 – 11111111 (8 bits)

• **HEXADECIMAL RANGE:** 00 – FF

DENARY RANGE: 0 – 255

1. BINARY TO HEXADECIMAL

Convert an 8-bit binary number into a 2-digit hexadecimal number.

- Split the 8-bit binary number into two groups of 4 bits from right to left.
- Convert each 4-bit group to its hexadecimal digit (using the table below).
- Write the hexadecimal digits from left to right.

EXAMPLE: $1101\ 0110_2$ to hexadecimal

- Step 1** Split into 4-bit groups:
- | | |
|--------|--------|
| 1101 | 0110 |
| D | 6 |
- Step 2** Convert each group:
- | | |
|------------|------------|
| $1101 = D$ | $0110 = 6$ |
|------------|------------|
- Step 3** Write the hex digits:
- $1101\ 0110_2 = D6_{16}$

2. HEXADECIMAL TO BINARY

Convert a 2-digit hexadecimal number into an 8-bit binary number.

- Replace each hexadecimal digit with its 4-bit binary equivalent (using the table below).
- Write the two 4-bit groups from left to right to make 8 bits.

EXAMPLE: $3A_{16}$ to binary

- Step 1** Convert each hex digit to 4-bit binary:
- | | |
|--------|--------|
| 3 | A |
| ↓ | ↓ |
| 0011 | 1010 |
- Step 2** Write the groups together:
- $0011\ 1010_2$

4-BIT BINARY ↔ HEXADECIMAL TABLE

Binary	0000	0001	0010	0011	0100	0101	0110	0111
Hex	0	1	2	3	4	5	6	7
Binary	1000	1001	1010	1011	1100	1101	1110	1111
Hex	8	9	A	B	C	D	E	F

PRACTICE: BINARY TO HEXADECIMAL

Convert the following 8-bit binary numbers to hexadecimal.

- $0000\ 1111_2 = \boxed{}_{16}$
- $1010\ 0011_2 = \boxed{}_{16}$
- $1111\ 0000_2 = \boxed{}_{16}$
- $0101\ 1101_2 = \boxed{}_{16}$
- $1100\ 1010_2 = \boxed{}_{16}$

PRACTICE: HEXADECIMAL TO BINARY

Convert the following hexadecimal numbers to 8-bit binary.

- $7C_{16} = \boxed{}_2$
- $2F_{16} = \boxed{}_2$
- $A5_{16} = \boxed{}_2$
- $01_{16} = \boxed{}_2$
- $FE_{16} = \boxed{}_2$

KEY POINTS



- ✓ Binary ↔ Hex conversions work in groups of 4 bits.
- ✓ 8 bits = 2 hex digits.
- ✓ Always split binary from right to left.
- ✓ Use leading zeros in binary if needed to make 8 bits.
- ✓ Binary range: 00000000 – 11111111
- ✓ Hex range: 00 – FF
- ✓ Denary range: 0 – 255



MEMORY TIP

Hex digits: 0 1 2 3 4 5 6 7 8 9 A B C D E F
A = 10, B = 11, C = 12, D = 13, E = 14, F = 15

Think: 4 bits = 1 hex digit
So 8 bits = 2 hex digits

DENARY & HEXADECIMAL CONVERSIONS

Convert positive denary whole numbers to 2-digit hexadecimal numbers and vice versa.

DENARY RANGE: 0 – 255

HEXADECIMAL RANGE: 00 – FF

(this is 2 digits in hexadecimal)

1. DENARY TO HEXADECIMAL

Convert a positive denary whole number (0–255) into a 2-digit hexadecimal number.

- 1 Divide the denary number by 16.
- 2 Write down the remainder (0–15).
If remainder is 10–15, use A–F.
- 3 Divide the quotient by 16.
- 4 Write down the next remainder.
- 5 The final quotient is the first (left) hex digit.
- 6 Write the remainders as hex digits (right digit first).

EXAMPLE: 175₁₀ to hexadecimal

Step	Calculation	Quotient	Remainder	Hex digit
1	175 ÷ 16	10	15	F
2	10 ÷ 16	0	10	A
3	0 ÷ 16	0	0	–

Read
bottom
to top

Hexadecimal: **AF₁₆**



The first (left) digit comes from the final quotient.
The second (right) digit comes from the first remainder.

2. HEXADECIMAL TO DENARY

Convert a 2-digit hexadecimal number into denary.

- 1 Write the place values:
Left digit × 16¹ + Right digit × 16⁰
- 2 Convert each hex digit to its denary value (A=10, B=11, ..., F=15).
- 3 Multiply and add the results.

EXAMPLE: 3C₁₆ to denary

Hex digit	3	C
Denary value	3	12
Place value	× 16 ¹ (16)	× 16 ⁰ (1)
Calculation	3 × 16 = 48	12 × 1 = 12

Add the results: 48 + 12 = 60

Denary: **60₁₀**

HEX DIGITS: VALUES AND PLACE VALUES

Hex digit	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Place values in hexadecimal		
																		16 ¹ (left digit)	16 ⁰ (right digit)
Denary value	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Denary	16	1
																	Example	3 × 16 = 48	C(12) × 1 = 12

DENARY TO HEXADECIMAL – QUICK REFERENCE

Denary	Hex	Denary	Hex
0	00	16	10
1	01	32	20
2	02	48	30
3	03	64	40
4	04	80	50
5	05	96	60
6	06	112	70
7	07	128	80
8	08	144	90
9	09	160	A0
10	0A	176	B0
11	0B	192	C0
12	0C	208	D0
13	0D	224	E0
14	0E	240	F0
15	0F	255	FF

PRACTICE: CONVERT THE FOLLOWING

A. Denary to hexadecimal (2 digits)

Convert these denary numbers to 2-digit hexadecimal.

- 1 25₁₀ = _____₁₆
- 2 67₁₀ = _____₁₆
- 3 128₁₀ = _____₁₆
- 4 200₁₀ = _____₁₆
- 5 255₁₀ = _____₁₆

B. Hexadecimal to denary

Convert these hexadecimal numbers to denary.

- 1 1A₁₆ = _____₁₀
- 2 3F₁₆ = _____₁₀
- 3 7C₁₆ = _____₁₀
- 4 A5₁₆ = _____₁₀
- 5 FF₁₆ = _____₁₀

KEY POINTS

- ✓ Hexadecimal is base 16.
- ✓ We use 0–9 for values 0–9.
- ✓ We use A–F for values 10–15.
- ✓ Two hexadecimal digits can represent 256 values (0–255).
- ✓ Left digit has a place value of 16.
- ✓ Right digit has a place value of 1.
- ✓ Binary, denary and hex are all different ways to represent the same numbers.

RANGES TO REMEMBER

Denary range: 0 – 255
Hexadecimal range: 00 – FF
Binary range: 0000000 – 11111111 (8 bits)



TIP: If the left digit is 0 (i.e. the number is less than 16), always write a leading zero.

Example: 5₁₀ = 05₁₆ | 15₁₀ = 0F₁₆

Need binary?

Convert via hex!

(Denary ↔ Hex ↔ Binary)

BINARY ADDITION, OVERFLOW AND SHIFTS

BINARY ADDITION RULES

1	$0 + 0 = 0$
2	$0 + 1 = 1$
3	$1 + 0 = 1$
4	$1 + 1 = 10$ – write 0, carry 1
5	$1 + 1 + 1 = 11$ – write 1, carry 1

WORKED BINARY ADDITION

$$\begin{array}{r}
 \begin{array}{ccccccc} 1 & 1 & 1 & & & & \end{array} \\
 00101101 \\
 + 00010111 \\
 \hline
 = 01000100
 \end{array}$$

$$45 + 23 = 68$$

WHAT IS OVERFLOW?

Overflow occurs when the result is too large to fit into the available number of bits.

$$\begin{array}{r}
 11110000 \\
 + 00100000 \\
 \hline
 \end{array}$$

 **1** 00010000

$$240 + 32 = 272$$

The result needs nine bits, so it cannot be stored as an eight-bit unsigned binary number.

BINARY SHIFTS

LEFT SHIFT

Original: 00110110 = 54

Shift left by 1: 01101100 = 108



Each bit moves one place to the left.



A zero fills the empty position on the right.



A one-place left shift normally multiplies a positive integer by 2.

Original: 11000001

← shift left by 1

Result: 10000010

leftmost 1
discarded

new 0
added

The lost bit does not reappear on the other side.

RIGHT SHIFT

Original: 00110110 = 54

Shift right by 1: 00011011 = 27



Each bit moves one place to the right.



A zero fills the empty position on the left.



A one-place right shift normally divides a positive integer by 2.

MULTIPLE PLACES



Left by 1
→ ×2



Left by 2
→ ×4



Left by 3
→ ×8



Right by 1
→ ÷2



Right by 2
→ ÷4



Right by 3
→ ÷8

Bits shifted beyond the available positions are lost.



EXAM TIP

A binary shift moves the bits and discards bits that leave the available positions. It does not rotate them.



Add with carries • overflow needs an extra bit • left grows • right reduces

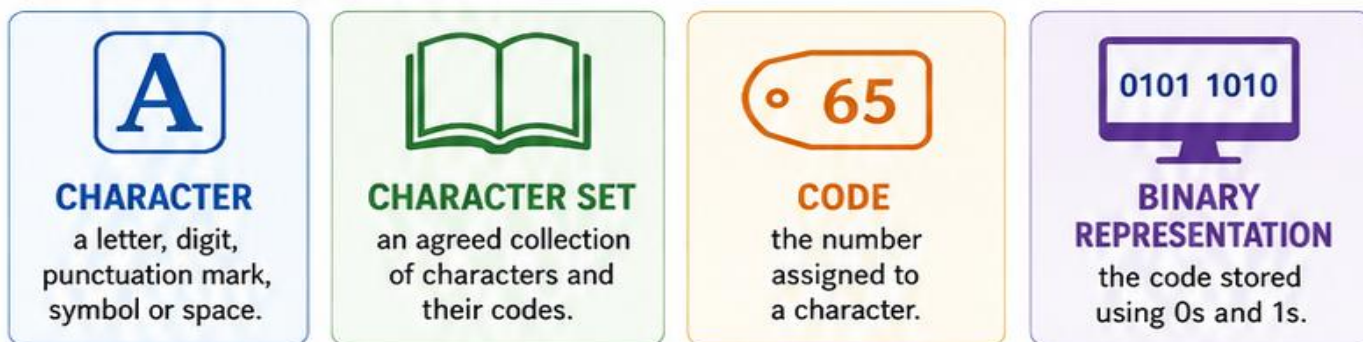
HOW COMPUTERS STORE CHARACTERS

CHARACTERS ARE STORED AS BINARY

A character set assigns a unique numerical code to each character.
The code is stored as binary.

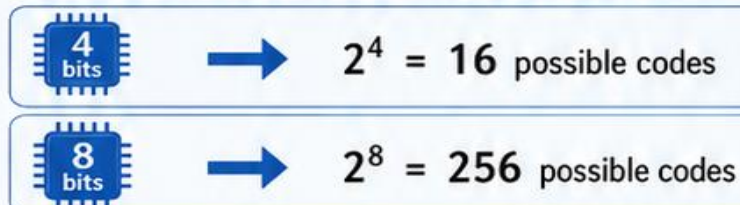


WHAT IS A CHARACTER SET?



BITS CONTROL THE NUMBER OF POSSIBLE CHARACTERS

Number of possible codes = 2^n , where n is the number of bits.



More bits per character allow a character set to represent more characters.

ASCII AND UNICODE



Unicode can represent far more characters than ASCII.



EXAM TIP ASCII and Unicode store codes representing characters.
They do not store pictures of the characters.

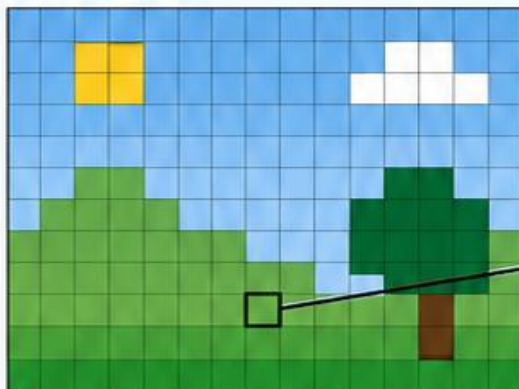
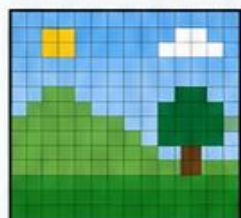


Character → code → binary • more bits → more possible characters

HOW COMPUTERS STORE IMAGES

BITMAP IMAGES ARE MADE FROM PIXELS

A bitmap image is a grid of pixels.
Each pixel has one colour represented by a binary code.



This pixel's colour is represented by the binary code:
01001101

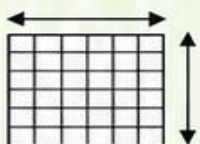
KEY IMAGE TERMS

PIXEL



the smallest individual element of a bitmap image.

RESOLUTION



the number of pixels used to represent an image, described by its width and height.

COLOUR DEPTH



the number of bits available to represent the colour of each pixel.

METADATA



additional information stored about the image, such as width, height, date created or camera settings.

COLOUR DEPTH AND POSSIBLE COLOURS

Number of possible colours = 2^n , where n is the colour depth.

1-bit colour	$2^1 = 2$		2 possible colours
2-bit colour	$2^2 = 4$		4 possible colours
8-bit colour	$2^8 = 256$		256 possible colours
24-bit colour	$2^{24} = 16,777,216$		16,777,216 possible colours

HOW CHANGES AFFECT QUALITY AND FILE SIZE

INCREASE RESOLUTION



- More pixels can show more detail.
- Image quality may improve.
- File size increases.

INCREASE COLOUR DEPTH



- More colours can be represented.
- Colour detail may improve.
- File size increases.



Increasing resolution or colour depth does not guarantee that the original image contains more useful detail.



EXAM TIP

Higher resolution and colour depth usually improve representation quality, but both increase the file size.



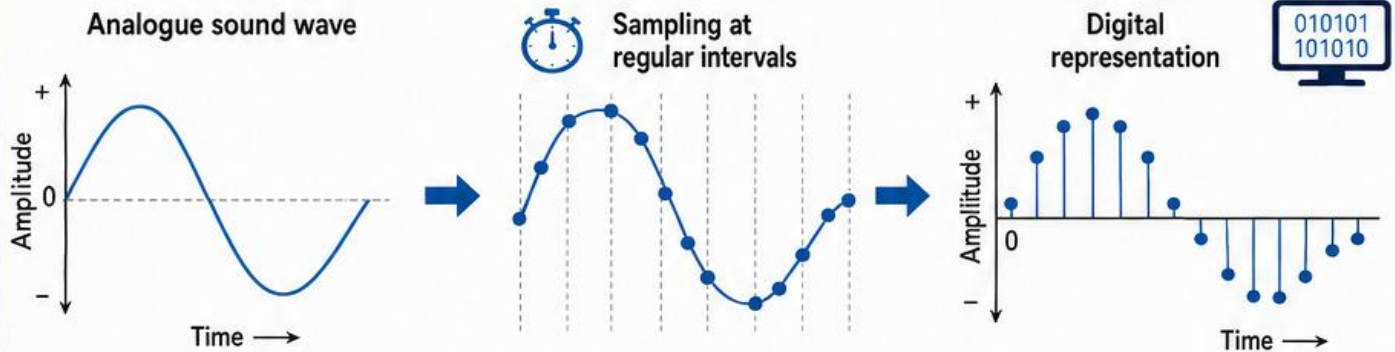
More pixels add detail • more bits add colours • both increase file size



HOW COMPUTERS STORE SOUND



Sampling measures the amplitude of a sound wave at regular intervals.



Sample rate

number of samples taken per second; measured in hertz (Hz).

Higher sample rate = more samples, more accurate sound, larger file.



Bit depth

number of bits used to store each sample.

Higher bit depth = more possible amplitude values, more accurate sound, larger file.



Duration

longer recordings contain more samples and create larger files.

FORMULA



Sound file size (bits) =

$$\text{sample rate (Hz)} \times \text{bit depth (bits)} \times \text{duration (seconds)}$$

For stereo, multiply by the number of channels.

WORKED EXAMPLE



$$20,000 \text{ Hz} \times 8 \text{ bits} \times 30 \text{ seconds} = 4,800,000 \text{ bits}$$

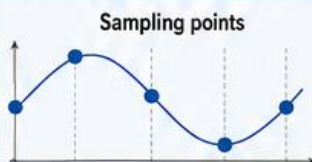
$$\div 8 = 600,000 \text{ bytes}$$

$$\div 1,000 = 600 \text{ KB}$$


Answer: 600 KB (using decimal conversions)

Effect of sample rate on how well the digital signal matches the original wave

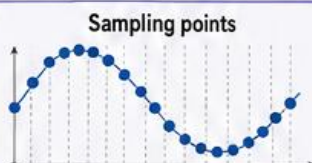
Low sample rate (fewer samples per second)



Result

Sparse samples miss detail. The digital signal is rough and does not follow the original wave closely.

High sample rate (more samples per second)



Result

More samples capture more detail. The digital signal follows the original wave much more closely.

EXAM TIP



Do not confuse sample rate with bit depth: rate controls how often; depth controls how precisely.

MEMORY HOOK



RATE = readings per second

• **DEPTH** = detail per reading

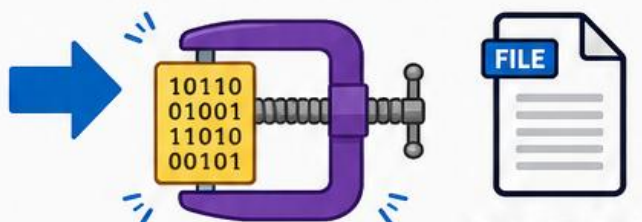


FILE COMPRESSION



LARGE FILE

Compression **reduces** the number of **bits** needed to store a file.



SMALLER FILE

WHY COMPRESS FILES?



uses less storage space



transfers and downloads faster



uses less bandwidth



Caution: compression does not automatically improve the quality of a file.

LOSSLESS COMPRESSION

- ✓ no data is permanently removed
- ✓ the original file can be reconstructed exactly
- ✓ usually produces a smaller reduction in file size than lossy compression
- ✓ suitable when every bit matters: text, program files and important data
- ✓ **examples:** PNG, ZIP, FLAC



TEXT



PROGRAM FILES



DATA

LOSSY COMPRESSION

- ✓ permanently removes some data
- ✓ the original file cannot be reconstructed exactly
- ✓ can greatly reduce file size
- ✓ quality may decrease, especially with repeated compression
- ✓ suitable for images, audio and video when some loss is acceptable
- ✓ **examples:** JPEG, MP3, MPEG



IMAGES



AUDIO



VIDEO

EXAMPLE: LOSSLESS (Run-Length Encoding)

A simple lossless method: repeated values are stored as a value and count.

A A A A A B B C C C



7A2B3C

(original data)

(compressed)

Not all lossless methods use run-length encoding.

EXAMPLE: LOSSY (Images / Audio)

Less noticeable data may be discarded to save space.

ORIGINAL (more detail)



LOSSY (less detail)



REMOVED
DETAIL



DECISION GUIDE



Choose **LOSSLESS** when exact recovery is essential.

Choose **LOSSY** when a much smaller file is more important than perfect quality.

EXAM TIP



Do not say lossless means 'no compression'. It compresses the file without permanently losing data.

MEMORY HOOK



LOSSLESS = lose no data

LOSSY = lose some data



MINI CHECK

Which type should store a program file?

Think it through! 🤔

NETWORKS & TOPOLOGIES

TYPES OF NETWORK

LAN (Local Area Network)

Connects devices in a small area such as a home, school or office.



Key features

- Small geographical area
- High data transfer speeds
- Owned and managed by one organisation

WAN (Wide Area Network)

Connects LANs across large geographical areas such as cities, countries or the world.



Key features

- Large geographical area
- Uses public or private connections
- Lower speeds than LANs
- More expensive

PAN (Personal Area Network)

Connects devices around a person, usually within a few metres.



Key features

- Very small range
- Low power
- Often wireless (e.g. Bluetooth)

COMMON NETWORK DEVICES

Switch



connects devices in a LAN and sends data to the correct device using MAC addresses.

Router



connects different networks together and routes data between them.

Access Point (Wireless AP)



allows wireless devices to connect to a wired network.

Modem



connects your network to the ISP via a phone line, cable or fibre.

Firewall



monitors and controls incoming and outgoing network traffic to keep networks secure.

IP ADDRESSES

Every device on a network has a unique IP address.



192.168.1.10

Network part

Network part

Network part

Device part

Key points

- IPv4 addresses use 4 numbers from 0–255 (e.g. 192.168.1.10)
- IPv6 uses longer addresses to allow more devices.
- Private IP ranges are used inside networks, public IPs on the Internet.

PROTOCOLS

Protocols are rules that govern how data is transmitted.



HTTP

Used to transfer web pages (not secure).



HTTPS

HTTP with encryption for secure web browsing.



SMTP

Used to send emails.



FTP

Used to transfer files between computers.



IMAP / POP3

Used to receive and manage emails.

TOPOLOGIES (NETWORK LAYOUTS)

Star



All devices connect to a central device. Easy to manage and faults are easy to find.

Bus



All devices connect to a single cable (backbone). Simple but a break in the cable affects all devices.

Ring



Each device connects to two others, forming a ring. Data travels in one or two directions.

Mesh



Devices connect to many others. Very reliable but expensive to set up.

Tree



A combination of star and bus layouts. Scalable and commonly used in organisations.

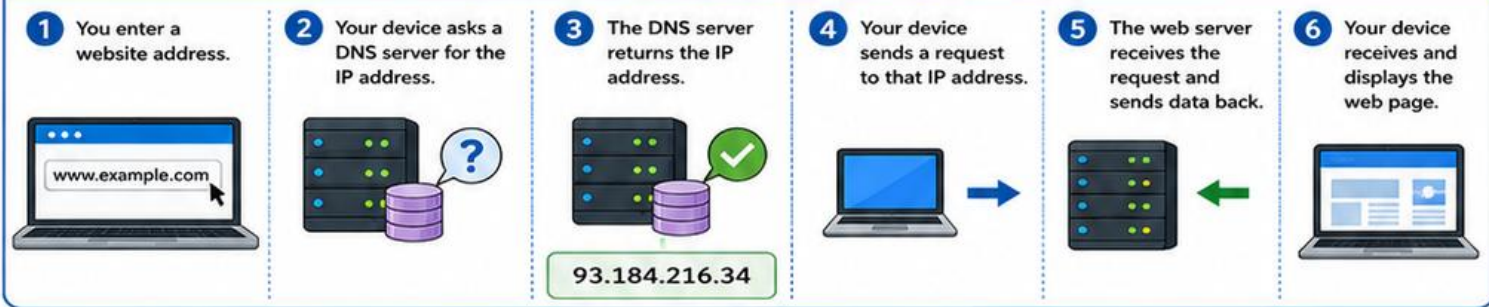


EXAM TIP: Be able to describe the advantages and disadvantages of each network type and topology. Think about cost, speed, reliability and security.



Networks connect devices.
Topologies are how the network is laid out.
Protocols are the rules that control communication.

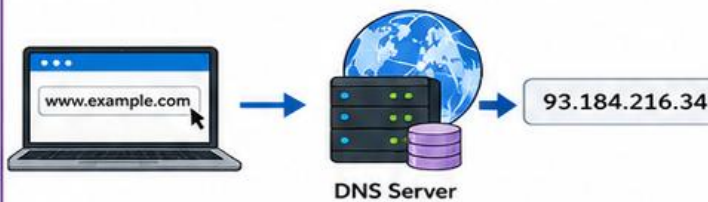
HOW THE INTERNET WORKS



EXAM TIP: The Internet is a 'network of networks'.
Data is split into packets, sent, then reassembled at the destination.

DNS: THE INTERNET'S ADDRESS BOOK

DNS (Domain Name System) converts easy-to-remember names into IP addresses.



Key points

- DNS makes the Internet user-friendly.
- Without DNS you would need to remember IP addresses.
- DNS works in a hierarchy from root servers down to local servers.
- The result is cached to make future lookups faster.

CLOUD COMPUTING

Cloud computing is delivering services (e.g. storage, software, processing power) over the Internet.



Examples of cloud services



Key points

- Data and services are stored on remote servers, not your device.
- Access from anywhere with an Internet connection.
- Scalable – resources can be increased or decreased as needed.
- Examples include storage, software as a service (SaaS) and platform as a service (PaaS).

BENEFITS OF THE INTERNET

- Access to information and services 24/7
- Communication and collaboration
- Online shopping and banking
- Education and online learning
- Cloud storage and backup
- Remote working opportunities

DRAWBACKS / RISKS

- Security threats (e.g. malware, hacking)
- Privacy risks and data collection
- Inappropriate content and misinformation
- Costs of Internet services and data
- Internet downtime and dependence
- Online scams and phishing

KEEPING SAFE ONLINE

It is important to protect data, devices and personal information when using the Internet.

- Use strong passwords and change them regularly.
- Keep software and antivirus up to date.
- Be careful with emails and links – don't click on anything suspicious.
- Do not share personal information online.
- Use secure connections (HTTPS). Look for the padlock in the address bar.
- Think before you post – once online, it can be hard to remove.

KEY POINTS SUMMARY

- ✓ **Networks** connect devices and can be LANs, WANs or PANs.
- ✓ Devices communicate using **IP addresses** and **protocols**.
- ✓ **DNS** converts domain names into IP addresses.
- ✓ The **Internet** works by requests and responses between clients and servers.
- ✓ **Cloud computing** provides services over the Internet, stored on remote servers.
- ✓ The Internet offers many **benefits** but also **risks**, so **staying safe** is essential.



EXAM TIP: Be able to describe, explain and give examples of all the key terms and technologies on these pages.

NETWORKS: KEY CONCEPTS

Performance, roles, hardware, servers & clients, and connection types

FACTORS THAT AFFECT THE PERFORMANCE OF NETWORKS

Several factors influence how well a network works.



Number of devices connected

More devices sharing the network can slow it down.



Bandwidth

The maximum amount of data that can be transferred. Higher bandwidth means faster data transfer.



Type of connection

Wired connections (Ethernet) are usually faster and more reliable than wireless.



Network traffic

Heavy data usage (e.g. streaming, downloading) can cause congestion and reduce speed.



Interference (wireless)

Walls, distance and other devices can weaken wireless signals.



Quality of network hardware

Outdated or low-quality hardware can reduce performance.

ROLES OF COMPUTERS IN NETWORKS

CLIENT-SERVER NETWORK

Clients request services or resources from a central server.



Server

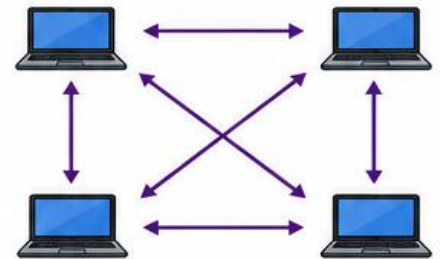
Provides and manages resources or services (e.g. files, web pages, printing) for clients.

Clients

Request and use resources or services from the server.

PEER-TO-PEER (P2P) NETWORK

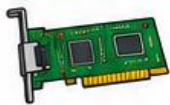
All computers (peers) are equal and can share resources directly.



- ✓ No central server.
- ✓ Each peer can act as both a client and a server.
- ✓ Simple and low cost to set up.
- ✓ Less secure and harder to manage on large networks.

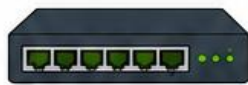
HARDWARE NEEDED TO CONNECT STAND-ALONE COMPUTERS INTO A LAN

NIC (Network Interface Controller/Card)



Installed in a computer. Allows it to connect to a network and provides a MAC address.

Switch



Connects devices in a LAN and sends data to the correct device using MAC addresses.

Wireless Access Point (WAP)



Allows wireless devices to connect to a wired network. Creates a Wi-Fi network.

Router



Connects different networks together and routes data between them (e.g. from a LAN to the Internet).

Transmission Media



Wired: Ethernet cable (twisted pair fibre)



Wireless: Uses radio waves

WEB SERVERS AND CLIENTS

Web Server



Stores websites (web pages) and other files. Responds to requests from clients and sends the requested data.

Web Client



Requests a web page from a web server using a web browser (e.g. Chrome, Edge).

Request (e.g. URL)

Response (web page/data)

THE INTERNET: A NETWORK OF NETWORKS



The Internet is a worldwide collection of many smaller networks connected together.



A Domain Name Service (DNS) is made up of many DNS servers located around the world.



DNS converts human-friendly URLs (e.g. www.example.com) into IP addresses (e.g. 93.184.216.34).

How DNS works



1 You enter a URL (e.g. www.example.com).



2 Your device asks a DNS server for the IP address.



3 The DNS server returns the IP address (e.g. 93.184.216.34).



4 Your device connects to the web server using the IP address.

1.3.2 WIRED AND WIRELESS NETWORKS, PROTOCOLS AND LAYERS – MODES OF CONNECTION

WIRED CONNECTIONS



- Uses cables (twisted pair copper or fibre).
- Very fast and reliable.
- Less affected by interference.
- Common in homes, schools and offices.

WIRELESS CONNECTIONS

Wi-Fi



- Uses radio waves.
- Connects devices to a wireless access point (router).
- Range affected by walls and distance.
- Common in homes, schools and public places.

Bluetooth



- Short-range wireless connection.
- Low power.
- Used for devices such as headphones, keyboards, phones and mice.
- Range typically up to ~10 m.

KEY POINTS SUMMARY

- ✓ Network performance depends on factors like number of devices and bandwidth.
- ✓ Client-server networks use a central server; peer-to-peer networks do not.
- ✓ Hardware such as switches, routers and access points connect computers into a LAN.
- ✓ Web servers store and deliver data; web clients request and use it.
- ✓ The Internet is a network of networks; DNS converts URLs to IP addresses.
- ✓ Networks can be wired (Ethernet) or wireless (Wi-Fi, Bluetooth), each with different characteristics.



EXAM TIP: Learn the roles of devices and the differences between network types. Be able to apply your knowledge to a scenario (e.g. choosing the best connection for a new office).

THREATS TO SYSTEMS AND NETWORKS

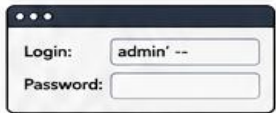


Cyber criminals use different forms of attack to try to gain unauthorised access to devices, networks or data, or to disrupt services.

KEY POINT

Attackers look for weaknesses (vulnerabilities) to achieve their goals. No system is 100% secure.

FORMS OF ATTACK

Attack	How the attack is used	Purpose of the attack	Example
Malware  Malicious software designed to cause harm or gain unauthorised access.	- Malware is installed on a device, often without the user's knowledge. - It can spread through infected files, email attachments, downloads or websites. - Runs in the background to carry out harmful actions.	 - Steal data or personal information. - Damage, corrupt or delete files. - Gain control of the device (e.g. as part of a botnet). - Monitor user activity (e.g. keylogging).	 A user opens an email attachment(s) to achieve a Trojan, giving the attacker remote access to the computer.
Social engineering (e.g. phishing)  Tricking people into revealing information or performing an action.	- Attackers pretend to be a trusted person or organisation. - Use emails, messages, phone calls or fake websites. - Create a sense of urgency, fear or curiosity to manipulate people.	 - Obtain sensitive information (e.g. passwords, bank details). - Gain unauthorised access to accounts or systems. - Deliver malware to a device.	 A fake email from a bank asks the user to "verify their account" by clicking a link to a fake website that steals their login details.
Brute-force attacks  Automatically trying many possible passwords or keys until the correct one is found.	- Software tries large numbers of passwords or key combinations very quickly. - Can target login pages, Wi-Fi networks or encrypted files. - Often uses automated tools.	 - Gain unauthorised access to accounts or systems. - Bypass security that relies on weak passwords. - Decrypt data or files.	 An attacker uses a tool to try thousands of passwords to break into a user account.
Denial of service (DoS) attacks  Overloading a system or network with traffic to make it unavailable to legitimate users.	- Attacker sends huge amounts of data or requests to a server, network or website. - Uses one device or many devices (botnet) to generate traffic. - Uses up the target's resources (bandwidth, memory, CPU).	 - Prevent legitimate users from accessing services. - Disrupt business and operations. - Cause financial or reputational damage.	 A website is flooded with traffic and becomes slow or goes offline, blocking real users.
Data interception and theft  Intercepting data as it is being transmitted and stealing it.	- Attacker captures data travelling over a network. - Often on unsecured (unencrypted) connections. - Tools called "packet sniffers" can be used.	 - Steal sensitive information (e.g. passwords, messages, credit card details). - Spy on communications. - Use stolen data for fraud or other attacks.	 On public Wi-Fi, an attacker intercepts unencrypted data containing a username and password.
SQL injection (concept)  Injecting malicious SQL code into input fields to manipulate a database.	- Attacker enters SQL commands into forms or URL parameters. - The application passes the input to the database without proper checking. - The injected code is executed by the database.	 - Access, change or delete data in the database. - Bypass login checks. - Gain full control of the website or application.	 An attacker enters malicious code in a login form and bypasses the login to access the system.

THREATS POSED TO DEVICES AND SYSTEMS

 Loss or theft of data (e.g. personal, financial, confidential).	 Disruption or unavailability of services and systems.	 Financial loss and business impact.	 Identity theft and fraud.	 Damage to reputation and loss of trust.	 Unauthorised access and loss of control.
--	--	--	--	--	---

CHECK YOUR UNDERSTANDING

- Which type of attack tricks people into giving away information?
 ?
- Which attack tries many passwords automatically until it finds the correct one?
 ?
- Which attack aims to make a service or website unavailable to users?
 ?
- Which attack involves intercepting data as it travels across a network?
 ?
- Which attack uses malicious SQL code in input fields?
 ?
- Give one threat posed by these attacks.
 ?



EXAM TIP: For each attack, remember: how it is used, the purpose of the attack and the threats it can cause.

PREVENTING VULNERABILITIES

How systems and networks are protected



Prevention methods help to reduce the risk of attacks by removing weaknesses (vulnerabilities) or limiting the impact when an attack occurs.

KEY POINT

No single method is 100% effective. Using multiple layers of protection (defence in depth) gives the best security.



COMMON PREVENTION METHODS

Prevention method	What it may limit / prevent	How it limits the attack	Related threats (from 1.4.1)
 Penetration testing	- Exploitation of unknown vulnerabilities. - Successful attacks by identifying weaknesses early.	- Ethical hackers simulate real attacks to find security weaknesses in systems or networks. - Issues are fixed before attackers can exploit them.	- Malware - Brute-force - DoS - SQL injection - Data interception
 Anti-malware software	- Malware infections (viruses, Trojans, spyware, ransomware). - Spread of malicious files.	- Scans files and downloads for known threats. - Detects and blocks or removes malware. - Often includes real-time protection.	Malware
 Firewalls	- Unauthorised access from outside networks. - Malicious network traffic.	- Monitors incoming and outgoing traffic. - Allows trusted traffic and blocks suspicious connections based on rules. - Acts as a barrier between networks.	- Malware - Brute-force - DoS - Data interception - SQL injection
 User access levels	- Unauthorised access to data and system functions. - Accidental or malicious changes to important data.	- Users are given only the access they need (least privilege). - Limits what users can view, change or delete.	- Malware - Brute-force - SQL injection - Data theft
 Passwords	- Unauthorised access to accounts and systems. - Identity theft.	- Strong passwords (long, complex, unique) are harder to guess or crack. - Prevents attackers from logging in as legitimate users.	- Brute-force - Social engineering - Data theft
 Encryption	- Data interception and theft. - Reading data without authorisation.	- Converts data into unreadable form using an algorithm and key. - Only someone with the correct key can decrypt and read the data.	Data interception and theft
 Physical security	- Unauthorised physical access to devices and systems. - Theft or tampering with hardware.	- Uses locks, alarms, CCTV, security guards and secure locations. - Stops attackers from gaining direct access to devices or stealing equipment.	- Malware (via USB) - Data theft - DoS (by tampering with hardware)

HOW THESE METHODS HELP AGAINST THREATS

 Against malware Anti-malware, firewalls and access levels block infections and limit the damage.	 Against social engineering Strong passwords, user education and access levels reduce the success of tricks like phishing.	 Against brute-force attacks Strong passwords, account lockout policies and firewalls limit repeated login attempts.	 Against DoS attacks Firewalls, intrusion detection and traffic filtering can block or reduce malicious floods of traffic.	 Against interception and theft Encryption protects data in transit; physical security and firewalls reduce the risk of eavesdropping.	 Against SQL injection Firewalls (web application firewalls) and secure coding prevent malicious SQL code from running.
--	---	---	---	---	--

CHECK YOUR UNDERSTANDING

1 Which prevention method would you use to remove vulnerabilities before an attacker finds them?  ?	2 Which method helps stop malware from infecting a system?  ?	3 Which method controls what a user is allowed to do on a system?  ?	4 Which method protects data if it is intercepted while travelling over a network?  ?	5 Which method would help stop an attacker from guessing passwords?  ?	6 Which method prevents unauthorised people from entering a server room?  ?
---	--	---	--	---	--



EXAM TIP:

When answering questions, always state the method, say what it prevents/limits, and explain how it works.

OPERATING SYSTEMS: PURPOSE & FUNCTIONALITY



An operating system (OS) is system software that manages computer hardware and software resources and provides common services for programs and users.

KEY POINT

The OS acts as an intermediary between the user and the hardware, ensuring the computer runs smoothly, securely and efficiently.



THE PURPOSE AND FUNCTIONALITY OF OPERATING SYSTEMS

1 USER INTERFACE



Allows users to interact with the computer and run applications.

Features of a user interface:

- ✓ Easy to use and intuitive
- ✓ Uses icons, menus, windows and buttons
- ✓ Supports mouse, keyboard and touch input
- ✓ Provides feedback (e.g. notifications, messages)
- ✓ Can be command-line (CLI) or graphical (GUI)

2 MEMORY MANAGEMENT & MULTITASKING



Manages the computer's main memory (RAM) and allows multiple programs to run at the same time.

Memory management:

- ✓ Transfers data between memory and the processor.
- ✓ Allocates memory to applications as needed.
- ✓ Keeps track of which parts of memory are free or in use.
- ✓ Moves data between primary storage and secondary storage if needed (e.g. virtual memory).

How it allows multitasking:

- ✓ Gives each running program a share of the CPU time (time slicing).
- ✓ Quickly switches between tasks so users can run multiple applications seemingly at once.

3 PERIPHERAL MANAGEMENT & DRIVERS



Controls and manages input, output and storage devices.

Includes:

- ✓ Communicating with peripherals (e.g. printer, keyboard, USB, monitor).
- ✓ Using device drivers – special software that allows the OS to talk to each device.
- ✓ Ensuring data is transferred correctly between devices and the processor.

Important:

Data is transferred between devices and the processor. This process needs to be managed by the OS to ensure everything works together without conflicts.

4 USER MANAGEMENT



Controls who can use the computer and what they can do.

Key functions:

- ✓ Allocation of an account for each user.
- ✓ Authentication – users must log in.
- ✓ Access rights – determines what each user can access or change (e.g. files, settings, programs).
- ✓ Security – protects data and systems from unauthorised access.
- ✓ Supports multiple users on the same system.

5 FILE MANAGEMENT



Organises, stores and manages files and folders on storage devices.

Key features:

- ✓ Naming – files and folders need unique names.
- ✓ Organising – creates and manages folders.
- ✓ Moving/Renaming – files can be moved, copied or renamed.
- ✓ Saving – data is written to storage.
- ✓ Deleting – removes files from storage (or sends to Recycle Bin).
- ✓ Searching – finds files quickly.
- ✓ Maintains file attributes (size, type, date, etc.).

HOW AN OPERATING SYSTEM WORKS

USERS & APPLICATIONS



Users interact with applications (e.g. web browser, games, word processor).

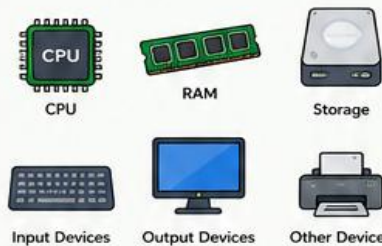
OPERATING SYSTEM (MANAGER)



The OS manages and coordinates all resources and activities.



HARDWARE RESOURCES



DATA FLOW

Data is input from a device
↓
The OS manages the transfer
↓
Data is processed by the CPU
↓
Results are output or saved

The OS ensures that resources are used efficiently, securely and fairly by all applications and users.

WHY THESE FUNCTIONS ARE IMPORTANT



Keeps the system running efficiently.



Protects data and prevents unauthorised access.



Allows multiple users and programs to use the system together.



Keeps files organised, easy to find and manage.



Ensures hardware and software work together smoothly.

CHECK YOUR UNDERSTANDING

1 What is the main purpose of an operating system?



2 Name two features of a graphical user interface.



3 How does memory management allow multitasking?



4 Why is data transfer between devices and the processor managed by the OS?



5 Name two functions of user management.



6 List two key features of file management.



EXAM TIP: Describe what the OS function does, give an example, and explain why it is important.

UTILITY SOFTWARE



Utility software are system programs that perform housekeeping and maintenance tasks. They help keep a computer secure, running efficiently and make better use of storage.

WHAT IS UTILITY SOFTWARE?

Utility software are programs that support the operating system by performing additional tasks that the operating system does not do by itself.

- ✓ Often included with the operating system or installed by the user
- ✓ Maintain, protect and optimise the computer system
- ✓ Improve performance, reliability and security
- ✓ Carry out tasks automatically or when needed



WHY IS UTILITY SOFTWARE NEEDED?

Operating systems manage files, memory and hardware, but they do not always include tools for these extra tasks.

Utility software:

- ✓ Keeps systems secure and protects data
- ✓ Helps maintain performance and stability
- ✓ Frees up storage space
- ✓ Prevents problems and data loss
- ✓ Allows computers to work at their best



TYPES OF UTILITY SYSTEM SOFTWARE

1. ENCRYPTION SOFTWARE

Purpose:

To protect data by converting it into a code so that only authorised users can read it.



How it works:

- Data is scrambled (encrypted) using an algorithm and a key.
- To read the data, the correct key is needed to decrypt it.

Why it is required:

- ✓ Protects sensitive or confidential information
- ✓ Prevents unauthorised access if data is stolen or lost
- ✓ Used for files, emails, discs, internet connections and full disk encryption

Examples:



BitLocker



VeraCrypt



PGP



7-Zip (AES)

2. DEFRAGMENTATION

Purpose:

To reorganise fragmented files on a storage device so data is stored in continuous blocks.



How it works:

- Over time, files become fragmented when they are saved, edited or deleted.
- Defragmentation rearranges the pieces so each file is stored in one continuous block.

Why it is required:

- ✓ Improves speed of accessing files
- ✓ Makes programs open and run faster
- ✓ Reduces wear on mechanical hard drives
- ✓ Helps the system run more efficiently

Examples:



Windows Defragmenter



Auslogics Disk Defrag



O&O Defrag

3. DATA COMPRESSION

Purpose:

To reduce the size of files or folders so they take up less storage space.



How it works:

- The software removes unnecessary or repeated data using compression algorithms.
- The file can be decompressed (unzipped) to restore the original data.

Why it is required:

- ✓ Saves storage space
- ✓ Faster to transfer files over networks or the internet
- ✓ Useful for backing up or archiving data
- ✓ Can group many files into one smaller file

Examples:



WinRAR



7-Zip



WinZip

COMPARISON OF UTILITY SOFTWARE

Utility Software	Main Purpose	Benefits	Example Tasks
 Encryption Software	Protects data by converting it into unreadable code.	✓ Keeps data secure and private ✓ Prevents data theft or misuse ✓ Essential for sensitive information	• Encrypt files or folders • Encrypt entire hard drives • Secure email and web connections
 Defragmentation	Reorganises fragmented files on storage devices.	✓ Improves system speed ✓ Faster file access and program loading ✓ Increases hard drive efficiency	• Analyse disk for fragmentation • Rearrange file fragments • Optimise storage performance
 Data Compression	Reduces file size by removing unnecessary data.	✓ Saves storage space ✓ Faster file transfers ✓ Easier to back up or share files	• Compress files or folders • Create ZIP/RAR archives • Extract (decompress) files

KEY POINTS TO REMEMBER



- ✓ Utility software performs housekeeping tasks that the operating system does not do.
- ✓ Encryption protects data by keeping it secret.
- ✓ Defragmentation speeds up access to files on storage devices.
- ✓ Data compression reduces file size to save space and speed up transfers.
- ✓ These tools help maintain security, speed, reliability and efficient use of resources.

UTILITY SOFTWARE IS NEEDED BECAUSE...



Improves Security



Improves Performance



Optimises Storage



Prevents Data Loss



Saves Time and Effort



Adds Extra Functionality

★ **EXAM TIP:** Be able to explain the purpose of each utility software type and give real-life examples of when they are used.

DIGITAL TECHNOLOGY & SOCIETY

Technology brings many benefits, but also important issues



Digital technology affects the way we live, work, learn and communicate.



KEY POINT

Technology introduces ethical, legal, cultural, environmental and privacy issues.

1. ETHICAL ISSUES



Questions about what is right or wrong.

EXAMPLES

- AI decision making
- Surveillance
- Addictive apps
- Digital addiction

POSITIVE IMPACT



Can improve lives and make things fairer.

NEGATIVE IMPACT



Unfair decisions, bias, harm to wellbeing.

2. LEGAL ISSUES



Laws that control how technology and data are used.

EXAMPLES

- Copyright
- Hacking
- Data protection
- Online safety laws

POSITIVE IMPACT



Protects rights and keeps people safe.

NEGATIVE IMPACT



Breaking laws can lead to fines or prison.

3. CULTURAL ISSUES



How technology affects beliefs, values and traditions.

EXAMPLES

- Globalisation
- Loss of traditions
- Digital divide
- Influence of media

POSITIVE IMPACT



Connects people and shares ideas.

NEGATIVE IMPACT



Loss of culture, misinformation, social division.

4. ENVIRONMENTAL ISSUES



The effect of technology on the environment.

EXAMPLES

- High energy use
- E-waste
- Resource use
- Carbon emissions

POSITIVE IMPACT



Supports efficiency and solutions to environmental problems.

NEGATIVE IMPACT



Pollution, e-waste and high energy consumption.

5. PRIVACY ISSUES



Concerns about personal data and how it is collected and used.

EXAMPLES

- Data collection
- Tracking
- Data breaches
- Identity theft

POSITIVE IMPACT



Personalised services and convenience.

NEGATIVE IMPACT



Loss of privacy, fraud and loss of trust.

EXAMPLES OF DIGITAL TECHNOLOGY AND THEIR IMPACT

	Social media	Smartphones	AI systems	Cloud storage	Online shopping
Ethical					
Legal	●	●	●	●	●
Cultural	●	●	●	●	●
Environmental	●	●	●	●	●
Privacy	●	●	●	●	●

- High impact
- Medium impact
- Low impact

BENEFITS



Convenience



Communication



Access to information



Innovation



Economic growth

WHAT CAN WE DO?



Use responsibly



Respect others



Protect personal data



Think sustainably



Know the laws

CHECK YOUR UNDERSTANDING

- Which issue is about what is right or wrong?
 ?
- Which issue is about laws and rules?
 ?
- Which issue is about traditions and beliefs?
 ?
- Which issue is about the environment?
 ?
- Which issue is about personal data and privacy?
 ?
- Give one example of technology and its impact.
 ?



EXAM TIP:

For impact questions, explain how technology affects society. Include both positive and negative points where relevant.

COMPUTER SCIENCE LEGISLATION

Laws that protect people, data, systems and original work



Legislation sets out rules for using computers and digital technology. Breaking these laws can lead to fines or imprisonment.



KEY POINT

These laws help to protect individuals, organisations and society by making sure technology is used responsibly and fairly.

1. DATA PROTECTION ACT 2018



Protects personal data and how organisations collect, store, use and share it.

✓ ALLOWS

- Organisations to collect and use personal data when there is a valid reason.
- People to see their own data.
- People to correct inaccurate data.
- People to request deletion of their data ("right to be forgotten").

✗ PROHIBITS

- Using personal data without lawful basis or consent.
- Failing to keep data secure.
- Sharing personal data without permission.
- Keeping data longer than necessary.

EXAMPLES

- Storing customer details without consent.
- Not securing a database containing personal information.
- Sending marketing emails without permission.

2. COMPUTER MISUSE ACT 1990



Protects computer systems from unauthorised access and misuse.

✓ ALLOWS

- Authorised users to access computer systems.
- System administrators to monitor and protect systems.
- Law enforcement to investigate computer crimes.

✗ PROHIBITS

- Unauthorised access to computer material or systems.
- Unauthorised access with intent to commit further offences.
- Unauthorised modification of data or programs.
- Making, supplying or using tools for hacking.

EXAMPLES

- Hacking into a company server.
- Accessing someone else's files without permission.
- Creating malware to attack systems.

3. COPYRIGHT DESIGNS AND PATENTS ACT 1988



Protects original works, designs, patents and intellectual property.

✓ ALLOWS

- Creators to control how their work is used.
- Users to use material with permission or under a licence.
- Fair dealing for research, private study, criticism or review.

✗ PROHIBITS

- Copying or sharing copyright material without permission.
- Distributing or selling unauthorised copies.
- Using copyrighted software, music, images, videos or text illegally.

EXAMPLES

- Downloading music or movies illegally.
- Copying text or images from a website without permission.
- Using software without a valid licence.

QUICK COMPARISON

LAW	MAIN PURPOSE	PROTECTS	✓ KEY ACTIONS ALLOWED	✗ KEY ACTIONS PROHIBITED
 Data Protection Act 2018	Protects personal data and how it is used.	Individuals' privacy and personal information.	- Collect and use data lawfully - Access and correct your data - Request deletion of your data	- Use data without consent - Fail to secure data - Share data without permission
 Computer Misuse Act 1990	Protects computer systems from unauthorised access and misuse.	Computer systems, data and programs from criminals.	- Access systems with permission - Monitor and protect systems - Investigate computer crimes	- Unauthorised access - Unauthorised data modification - Use of hacking tools
 Copyright Designs and Patents Act 1988	Protects original creative works and intellectual property.	Creators' rights and their original work.	- Control how work is used - Use with permission or licence - Fair dealing for study or review	- Copy without permission - Distribute unauthorised copies - Use without a licence

CHECK YOUR UNDERSTANDING

- 1 What is the main purpose of the Data Protection Act 2018?



?

- 2 Name two actions that are prohibited by the Computer Misuse Act 1990.



?

?

- 3 What does the Copyright Designs and Patents Act 1988 protect?



?



EXAM TIP:

Give a clear explanation of what each law does and examples of actions that are allowed and prohibited.

SOFTWARE LICENCES

Control how software can be used, changed and shared



A software licence is a legal agreement between the creator (or owner) of the software and the user. It sets out what you can and cannot do with the software.



KEY POINT

Licences protect the rights of the creator and help users understand how they can legally use software.

TWO MAIN TYPES OF SOFTWARE LICENCES

OPEN SOURCE LICENCES



Open source software gives users access to the source code. Users are allowed to use, change and share the software.

FEATURES OF OPEN SOURCE



Access to source code

Anyone can view how the software works.



Can be modified

Users can change the code to improve it or add new features.



Can be shared

Users can share the software with others, often under the same open licence.



Often free

Most open source software is free to use.



Transparent and trustworthy

The code can be checked for security flaws and improved by the community.



Community driven

Developed and improved by a community of volunteers and organisations.

EXAMPLES



Linux



Firefox



LibreOffice



GIMP



Blender



VS Code

PROPRIETARY (CLOSED SOURCE) LICENCES



Proprietary software does not give access to the source code. Users can use the software but must follow the licence terms set by the owner.

FEATURES OF PROPRIETARY



No access to source code

Users cannot see or change how the software works.



Restricted use

Users must follow the licence terms set by the owner.



Cannot be freely shared

Sharing or copying may be limited or not allowed.



Usually paid

Typically bought once or as a subscription.



Controlled by the company

Updates and improvements are decided by the owner.



Official support

Users usually get customer support from the company.

EXAMPLES



Windows



Microsoft Office



Adobe Photoshop



AutoCAD



iOS

OPEN SOURCE vs PROPRIETARY

ASPECT	OPEN SOURCE	PROPRIETARY
Source code	Available to view and modify	Not available
Modification	Allowed	Not allowed (only by the owner)
Cost	Usually free	Paid (one-off or subscription)
Sharing	Allowed (under the same licence)	Restricted by the licence
Support	Community support	Official support from the company
Best for	Customisation, transparency, learning, low cost	Ease of use, official support, reliability, polished products

SUMMARY

- ✓ Open source = open, editable, shareable, community driven.
- ✓ Proprietary = closed, controlled by the owner, usually paid.
- ✓ Choose the licence that best suits your needs and the task.

CHECK YOUR UNDERSTANDING

- What is the main feature of open source software?
 ?
- Name one benefit of open source software.
 ?
- What is the main feature of proprietary software?
 ?
- Why is proprietary software usually paid?
 ?
- Give one example of open source software.
 ?
- Give one example of proprietary software.
 ?



EXAM TIP:

When answering questions, give the purpose of each type of licence and explain one benefit and one drawback where relevant.