Cambridge (CIE) AS Computer Science

Your notes

Data Representation

Contents

- Binary prefixes
- Number systems
- Binary arithmetic
- Uses of number systems
- Character encoding

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Binary prefixes

What is a binary prefix?

- A binary prefix is a unit prefix used to **indicate multiples of bytes in binary**
- Consider the word **kilobyte**, "kilo" is the **prefix**
- **Bytes** are the smallest unit of data that can be stored in a computer so there needs to be a way of expressing bytes in larger multiples

Denary prefixes

- A common way of expressing multiples of bytes is to use denary prefixes:

Denary unit	Equivalent size (bytes)
1 kilobyte (1 KB)	1000
1 megabyte (1 MB)	1,000,000
1 gigabyte (1 GB)	1,000,000,000
1 terabyte (1 TB)	1,000,000,000,000
1 petabyte (1 PB)	1,000,000,000,000,000

- This system relies on the assumption that **1 kilo = 1000**
- This assumption is based on the **denary (base 10)** number system
  - **E.g. a 1 GB hard drive can store 1 x 10^9 bytes**

Binary prefixes

- However, computers use the **binary (base 2)** number system so the denary system is technically inaccurate when describing storage
- To be precise, expressing multiples of bytes is done using binary prefixes:

Binary unit	Number of bytes (base 2)	Equivalent size (base 10)
1 kibibyte (1 KiB)	2^{10}	1024
1 mebibyte (1 MiB)	2^{20}	1,048,576

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

1 gibibyte (1 GiB)	2^{30}	1,073,741,824
1 tebibyte (1 TiB)	2^{40}	1,099,511,627,776
1 pebibyte (1 PiB)	2^{50}	1,125,899,906,842,624

- Notice the prefixes change depending on the system being used, e.g. **kilo** (denary) vs **kibi** (binary)

Why does it matter?

- The importance of the system being used depends on how precise you need to be
- Identifying the **total amount of memory (RAM)** available to a computer must be accurate (use binary prefixes)
  - **E.g. 16 GiB RAM can store 16 x 2^{30} bytes of data (17,179,869,184 bytes)**
- when describing **storage space**, a rough estimate is acceptable (use denary prefixes)
  - **E.g. a 16 GB memory stick can store 16 x 10^9 bytes of data (16,000,000,000 bytes)**



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

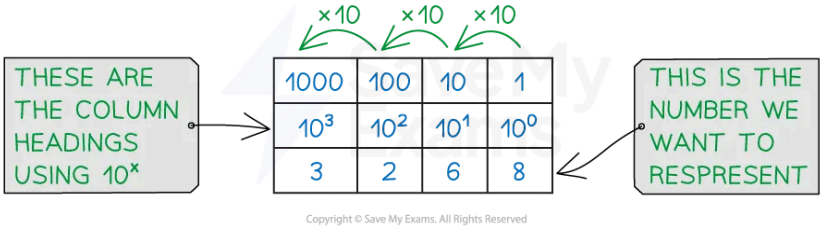
Number bases

What is a number base?

- A number base is the number of **different digits or symbols** a number system uses to represent values
- Each place in a number represents a **power of the base**, starting from the right

Denary

- Denary is a number system that is made up of **10 digits (0–9)**
- Denary is referred to as a **base-10** number system
- Each digit has a weight factor of **10 raised to a power**, the rightmost digit is 1s (10^0), the next digit to the left 10s (10^1) and so on
- Humans use the denary system for **counting**, **measuring** and **performing maths calculations**
- Using combinations of the 10 digits we can represent any number



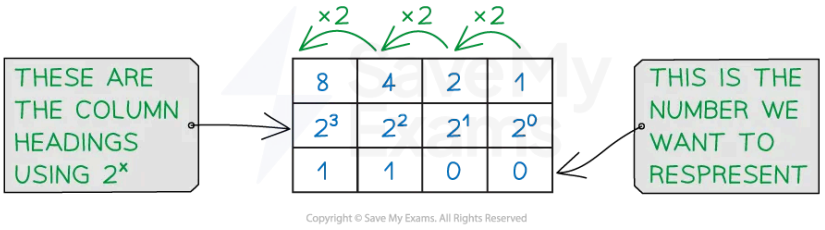
Copyright © Save My Exams. All Rights Reserved

- In this example, (3 x 1000) + (2 x 100) + (6 x 10) + (8 x 1) = 3268
- To represent a bigger number we **add more digits**

Binary

- Binary is a number system that is made up of **two digits (1 and 0)**
- Binary is referred to as a **base-2** number system
- Each digit has a weight factor of **2 raised to a power**, the rightmost digit is 1s (2^0), the next digit to the left 2s (2^1) and so on
- Each time a new digit is added, the **column value is multiplied by 2**
- Using **combinations of the 2 digits** we can represent any number

Study copy distributed by EduResource ZA | eduresourcza.com



Your notes

- In this example, **Binary 1100 = (1 x 8) + (1 x 4) = 12**
- To represent bigger numbers we add more **b**inary dig**its** (bits)

32,768	**16,384**	**8,192**	**4,096**	**2,048**	**1,024**	**512**	**256**	**128**	**64**	**32**	**16**	**8**	**4**
2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}

Hexadecimal

- Hexadecimal is a number system that is made up of **16 digits**, 10 numbers (**0–9**) and 6 letters (**A–F**)

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

- Hexadecimal is referred to as a **base-16** number system
- Each digit has a weight factor of **16 raised to a power**, the rightmost digit is 1s (16^0), the next digit to the left 16s (16^1)

16s	**1s**	
1	**3**	
1 x 16	3 x 1	= 19

- A quick comparison table demonstrates a **relationship** between **hexadecimal and a binary nibble**
- One hexadecimal digit can represent **four bits** of binary data

Denary	**Binary**	**Hexadecimal**
0	**0000**	**0**

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

1	**0001**	**1**
2	**0010**	**2**
3	**0011**	**3**
4	**0100**	**4**
5	**0101**	**5**
6	**0110**	**6**
7	**0111**	**7**
8	**1000**	**8**
9	**1001**	**9**
10	**1010**	**A**
11	**1011**	**B**
12	**1100**	**C**
13	**1101**	**D**
14	**1110**	**E**
15	**1111**	**F**

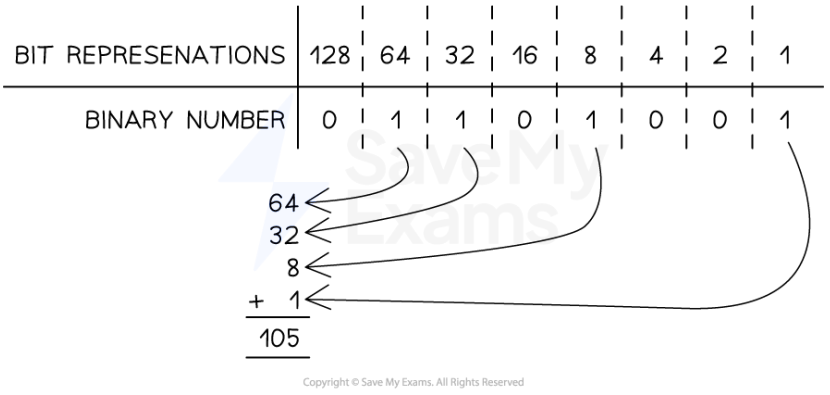


Your notes

Binary to denary & denary to binary

- Binary numbers can be converted into denary and vice-versa
- For example the 8-bit binary number **01101001** can be converted into denary using the following method:

Study copy distributed by EduResource ZA | eduresourcza.com



Copyright © Save My Exams. All Rights Reserved

Binary to denary conversion

Your notes

- Therefore the 8–bit binary number **01101001** is **105** as a denary value
- To convert the denary number **101** to binary, we firstly write out binary number system

128	64	32	16	8	4	2	1

- Then we start at the left and look for the highest number that is less than or equal to **101** and if so, place a 1 in that column
- Otherwise, place a 0 in the column
- **128** is bigger than **101** and therefore we place a 0 in that column
- **64** is smaller than **101** so we place a 1 in that column
  - **101 – 64 = 37**
  - This now means we have **37** left to find
- **32** is smaller than **37** so we place a 1 in that column
  - **37 – 32 = 5**
  - This now means we have **5** left to find
- **16** is bigger than **5** and therefore we place a 0 in that column
- **8** is bigger than **5** and therefore we place a 0 in that column
- **4** is smaller than **5** so we place a **1** in that column
  - **5 – 1 = 1**
  - This now means we have **1** left to find
- **2** is bigger than **1** and therefore we place a 0 in that column
- **1** is equal to the number we have left so we place a 1 in that column

Study copy distributed by EduResource ZA | eduresourcza.com

- 64 + 32 + 4 + 1 = **101**
  - Therefore the denary number **101** in binary is **01100101**

128	64	32	16	8	4	2	1
0	1	1	0	0	1	0	1



Your notes

Hexadecimal to denary & denary to hexadecimal

- To convert the denary number **163** to hexadecimal, start by **dividing the denary value by 16** and recording the **whole times** the number goes in and the **remainder**
- 163 ≡16 = 10 remainder 3
- In hexadecimal the **whole number = digit 1** and the **remainder = digit 2**
- Digit 1 = 10 (A)
- Digit 2 = 3
- Denary **163** is **A3** in hexadecimal
- To convert the hexadecimal number **79** to denary, start by **multiplying the first hexadecimal digit by 16**
- 7 ✖ 16 = 112
- **Add digit 2** to the result
- 112 + 9 = 121
- Hexadecimal **79** is **121** in denary

Binary to hexadecimal & hexadecimal to binary

- To convert the binary number **10110111** to **hexadecimal**, first split the 8 bit number into 2 binary **nibbles**

8	4	2	1		8	4	2	1
1	0	1	1		0	1	1	1

- For each nibble, convert the binary to it's denary value
- (1 x 8) + (1 x 2) + (1 x 1) = **11 (B)**
- (1 x 4) + (1 x 2) + (1 x 1) = **7**
- Join them together to make a 2 digit hexadecimal number
- Binary **10110111** is **B7** in hexadecimal
- To convert the hexadecimal number **5F** to binary, first **split the digits apart** and convert each to a **binary nibble**

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

8	4	2	1	
0	1	0	1	= 5

8	4	2	1	
1	1	1	1	= 15 (F)

- Join the 2 binary nibbles together to create an 8 bit binary number

128	64	32	16	8	4	2	1
0	1	0	1	1	1	1	1

- Hexadecimal **5F** is **01011111** in binary

Worked Example

Convert the positive binary integer 1010 0010 into hexadecimal. **[1]**

Answer

- A2 [1 mark]

Binary Coded Decimal (BCD)

What is Binary Coded Decimal?

- Binary coded decimal is a number system that uses **4 bit codes to represent each denary digit** (0–9)

BCD	Denary digit
0000	0
0001	1
0010	2
0011	3
0100	4

Your notes

© 2026 Save My Exams, Ltd. | Get more and ace your exams at savemyexams.com

Study copy distributed by EduResource ZA | eduresourcza.com

0101	5
0110	6
0111	7
1000	8
1001	9


Your notes

- To represent the denary number **2500** in **BCD** format would be:
  - **2500** = 0010 0101 0000 0000
- BCD can be stored in a computer as **either half a byte** (4 bits) or as **two 4 bit codes** joined to from a byte (8 bits)

Four single bytes

0	0	0	0	0	0	1	0	**= 2**
0	0	0	0	0	1	0	1	**= 5**
0	0	0	0	0	0	0	0	**= 0**
0	0	0	0	0	0	0	0	**= 0**

Two bytes

0	0	1	0	0	1	0	1	**= 2, 5**
0	0	0	0	0	0	0	0	**= 0, 0**

One's & two's complement

What is one's complement?

- One's complement is a method of representing **both positive and negative numbers**
- To turn a positive binary number in to a negative the **positive binary number is inverted** (0 becomes 1 and 1 becomes 0)

0	1	0	0	1	0	0	0	**= 72**

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

1	0	1	1	0	1	1	1	= -72

- The issue with one's complement is that it can have **two representations for 0** (positive and negative)



Examiner Tips and Tricks

Be careful not to confuse one's complement with two's complement. In one's complement, you just invert the bits to get the negative version, you don't treat the MSB as negative like you would in two's complement. That's a different system!

What is two's complement?

- Two's complement is another method of representing **both positive and negative numbers**
- To turn a positive binary number into a negative:
  - Positive binary number is **inverted**
  - **'1' is added** to the right most bit
- Using two's complement the **leftmost bit** is designated the **most significant bit** (**MSB**)
- To represent negative numbers this bit **must equal 1**, turning the column value into a **negative**
- Working with **8 bits**, the **128** column becomes **-128**

-128	**64**	**32**	**16**	**8**	**4**	**2**	**1**	
0	1	0	0	1	0	0	0	**= 72**
invert								
1	0	1	1	0	1	1	1	
add 1								
							1	**+**
result								
1	0	1	1	1	0	0	0	**= -72**



Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

- **8 bit two's complement** can represent values between 0111 1111 (**+127**) and 1000 0000 (**–128**)
- An alternative way of thinking about this is:
  - Starting from the right, keep all the bits the same **up to and including** the first 1
  - Flip the rest of the digits
    - 0101 1000 becomes...
    - 1010 1000



Your notes

Key reasons to use two's complement

- **Consistency**
  - You **don't need** different rules for signed and unsigned numbers
  - Whether you're adding +37 and +58, or –37 and +58, or –37 and –58, the **binary addition process is the same**
  - Two's complement works for **any combination** of positive and negative values
- **Hardware simplicity**
  - **CPUs** are built to do simple binary addition
  - Two's complement means:
    - The same **adder circuit** can handle everything
    - No special logic is needed to detect signs or subtract manually
- **It still works, up to a point**
  - As long as the **sum doesn't exceed the max value (+127 in 8-bit)**, you can add two positive numbers with two's complement without any issues



Worked Example

Convert the denary number 649 into Binary Coded Decimal (BCD) **[1]**

Answer

- 0110 0100 1001 [1 mark]

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

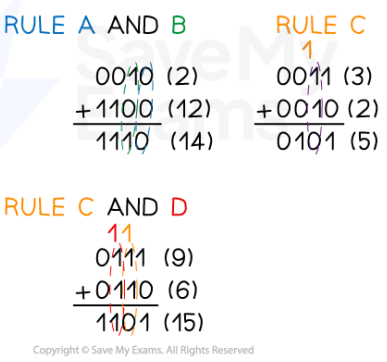
Binary addition

What is binary addition?

- Binary addition involves **summing numbers in base-2**, which uses only the digits 0 and 1
- Like denary addition, start from the **rightmost digit** and move towards the left
- **Carrying over** occurs when the sum of a column is greater than 1, passing the excess to the next left column

Example addition

RULE A: 0 + 0 = 0
RULE B: 0 + 1 = 1
RULE C: 1 + 1 = 0 CARRY 1
RULE D: 1 + 1 + 1 = 1 CARRY 1



Copyright © Save My Exams. All Rights Reserved

Worked Example

Add the following two binary integers using binary addition. Show your working. **[2]**

10011010 + 00101111

Answer

carry		1	1	1	1	1		
	1	0	0	1	1	0	1	0
+	0	0	1	0	1	1	1	1
	1	1	0	0	1	0	0	1

Study copy distributed by EduResource ZA | eduresourcza.com

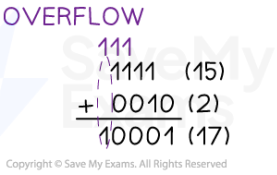
- 1 mark for working [1 mark]
- 1 mark for answer [1 mark]

Your notes

Overflow

What is an overflow?

- Overflow occurs when the sum of two binary numbers **exceeds the given number of bits**
- In **signed number** representations, the leftmost bit often serves as the sign bit; overflow can flip this, incorrectly changing the sign of the result
- Overflow generally leads to **incorrect or unpredictable results** as the extra bits are truncated or wrapped around



Binary subtraction

- To carry out subtraction, the number being subtracted is **converted into its negative equivalent** using two's complement
- The two numbers are then **added together**

Example

- 48 - 12

	-128	64	32	16	8	4	2	1	
	0	0	1	1	0	0	0	0	= 48
	0	0	0	0	1	1	0	0	= 12
	find two's complement of -12								
	0	0	0	0	1	1	0	0	= 12
	invert								
	1	1	1	1	0	0	1	1	

Study copy distributed by EduResource ZA | eduresourcza.com

	add 1								
								1	
	result								
	1	1	1	1	0	1	0	0	**= -12**
	Add 48 and -12								
	-128	**64**	**32**	**16**	**8**	**4**	**2**	**1**	
	0	0	1	1	0	0	0	0	**= 48**
	1	1	1	1	0	1	0	0	**= -12**
	carries								
1	1	1	1						
	result								
	0	0	1	0	0	1	0	0	**= 36**

- 48 - 12 = **36**
- The **additional overflow bit is ignored** leaving a result of 0010 0100, denary equivalent of **36**, which is the **correct answer**
- In **two's complement** arithmetic, the **overflow bit** does not contribute to the actual value of the operation but is more of a **by-product** of the method

Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com



Your notes

Applications of Binary Coded Decimal (BCD)

What are the uses of binary-coded decimal (BCD)?

Use Case	Why BCD is used
Electronic calculators	Keeps numbers in decimal format for easier display and accuracy
Digital clocks and watches	Time is naturally decimal (e.g. 12:45), so BCD makes display logic simpler
Banking and financial systems	Avoids rounding errors when doing decimal calculations, especially with money
Old digital systems / embedded systems	Simpler to implement with hardware that displays digits individually

- Binary coded decimal is commonly used in systems that need to **display decimal numbers clearly** and accurately
- BCD is ideal for applications like digital clocks, calculators, and financial systems **where decimal precision matters**
- Using BCD **avoids rounding errors** that can occur in binary-based arithmetic, especially with money and time
- It's still found in older or embedded systems where **simple hardware-based decimal** output is needed

Applications of hexadecimal

Why is hexadecimal used?

- In Computer Science hexadecimal is often preferred when working with **large values**
- It takes **fewer digits to represent a given value** in hexadecimal than in binary
  - 1 hexadecimal digit corresponds 4 bits and can represent 16 unique values (0–F)
- It is **beneficial** to use hexadecimal over binary because:
  - The more bits there are in a binary number, the **harder it makes for a human to read**
  - Numbers with more bits are more **prone to errors** when being copied

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

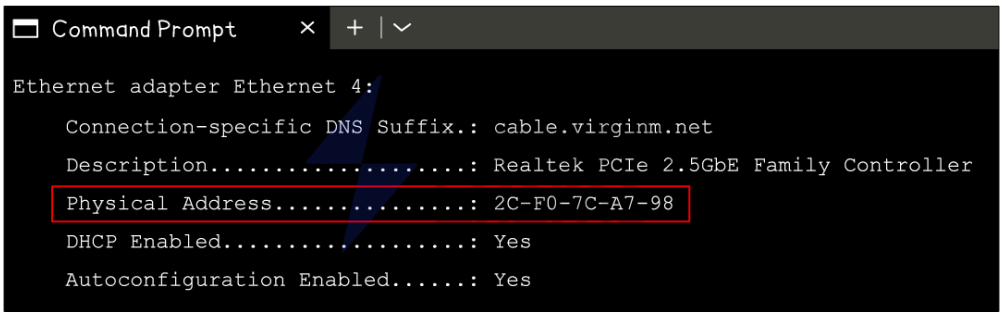
- Examples of where hexadecimal can be seen:
  - **MAC addresses**
  - **Colour codes**
  - **URLs**



Your notes

MAC addresses

- A typical MAC address consists of **12 hexadecimal digits**, equivalent to **48 digits** in binary
  - **AA:BB:CC:DD:EE:FF**
  - **10101010:10111011:11001100:11011101:11101110:11111111**
- **Writing down** or **performing calculations** with 48 binary digits makes it **very easy to make a mistake**

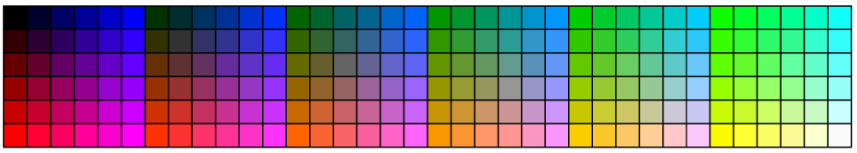


Copyright © Save My Exams. All Rights Reserved

Colour codes

- A typical hexadecimal colour code consists of **6 hexadecimal digits**, equivalent to **24 digits** in binary
  - **#66FF33** (green)
  - **01000010:11111111:00110011**

Study copy distributed by EduResource ZA | eduresourcza.com



51.102.255 #3366FF	102.51.255 #6633FF	204.51.255 #CC33FF	255.51.204 #FF33CC
51.204.255 #33CCFF	0.61.245 #003DF5	0.46.184 #002EB8	255.51.102 #FF3366
51.255.204 #33FFCC	184.138.0 #B88A00	245.184.0 #F5B800	255.102.51 #FF6633
51.255.102 #33FF66	102.255.51 #66FF33	204.255.51 #CCFF33	255.204.51 #FFCC33

Your notes

URL's

- A URL can only contain **standard characters** (a-z and A-Z), **numbers** (0-9) and some **special symbols** which is enough for basic web browsing
- If a URL needs to include a character **outside of this set**, they are converted into a **hexadecimal code**
- Hexadecimal codes included in a URL are prefixed with a **%** sign

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Character sets

What is a character set?

- A character set is **all the characters and symbols** that can be **represented** by a computer system
- Each character is given a **unique** binary code
- Character sets are **ordered logically**, the code for 'B' is one more than the code for 'A'
- A character set provides a **standard** for computers to communicate and send/receive information
- Without a character set, one system might interpret 01000001 differently from another
- The number of characters that can be represented is determined by the **number of bits used** by the character set
- Two common character sets are:
  - American Standard Code for Information Interchange (**ASCII**)
  - Universal Character Encoding (**UNICODE**)

ASCII

What is ASCII?

- ASCII is a character set and was an accepted standard for information interchange
- ASCII uses **7 bits**, providing 2^7 unique codes (128) or a maximum of **128 characters** it can represent
- This is enough to represent the letters, numbers, and symbols from a **standard keyboard**
- The **sixth bit** changes from 1 to 0 when **comparing uppercase and lowercase characters**
  - **a** 0110 0001
  - **A** 0100 0001
  - **b** 0110 0010
  - **B** 0100 0010
- This made conversion between the two much easier
- This speeds up the overall usability of the character set

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Letter	ASCII Code	Binary	Letter	ASCII Code	Binary
a	097	01100001	A	065	01000001
b	098	01100010	B	066	01000010
c	099	01100011	C	067	01000011
d	100	01100100	D	068	01000100
e	101	01100101	E	069	01000101
f	102	01100110	F	070	01000110
g	103	01100111	G	071	01000111
h	104	01101000	H	072	01001000
i	105	01101001	I	073	01001001
j	106	01101010	J	074	01001010
k	107	01101011	K	075	01001011
l	108	01101100	L	076	01001100
m	109	01101101	M	077	01001101
n	110	01101110	N	078	01001110
o	111	01101111	O	079	01001111
p	112	01110000	P	080	01010000
q	113	01110001	Q	081	01010001
r	114	01110010	R	082	01010010
s	115	01110011	S	083	01010011
t	116	01110100	T	084	01010100
u	117	01110101	U	085	01010101
v	118	01110110	V	086	01010110
w	119	01110111	W	087	01010111
x	120	01111000	X	088	01011000
y	121	01111001	Y	089	01011001
z	122	01111010	Z	090	01011010

Copyright © Save My Exams. All Rights Reserved

Part of the ASCII code table

Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

- ASCII **only** represents basic characters needed for **English**, limiting its use for other languages



Your notes

Extended ASCII

What is extended ASCII?

- Extended ASCII uses an **8th bit**, providing 256 unique codes ($2^8 = 256$) or a maximum of **256 characters** it can represent
- Extended ASCII provides **essential characters** such as mathematical operators and more recent symbols such as ©
- This allows for **non-English characters** and for drawing characters to be included

UNICODE

What is UNICODE?

- UNICODE is a **character set** and was created as a solution to the limitations of ASCII
- UNICODE uses a minimum of **16 bits**, providing 2^{16} unique codes (65,536) or a minimum of **65,536 characters** it can represent
- UNICODE can represent characters from **all the major languages** around the world
- UNICODE was designed to **create a universal standard** that covered all languages and all writing systems
- The **first 128 characters** in the UNICODE character set are the same as ASCII

ASCII vs UNICODE

	ASCII	UNICODE
Number of bits	7-bits	16-bits
Number of characters	128 characters	65,536 characters
Uses	Used to represent characters in the English language.	Used to represent characters across the world.
Benefits	It uses a lot less storage space than UNICODE.	It can represent more characters than ASCII. It can support all common characters across the world. It can represent special characters such as emoji's.

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Drawbacks	It can only represent 128 characters. It cannot store special characters such as emoji's.	It uses a lot more storage space than ASCII.



Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Multimedia

Contents

- Bitmap images
- Vector graphics
- Sound representation

© 2026 Save My Exams, Ltd. | Get more and ace your exams at savemyexams.com

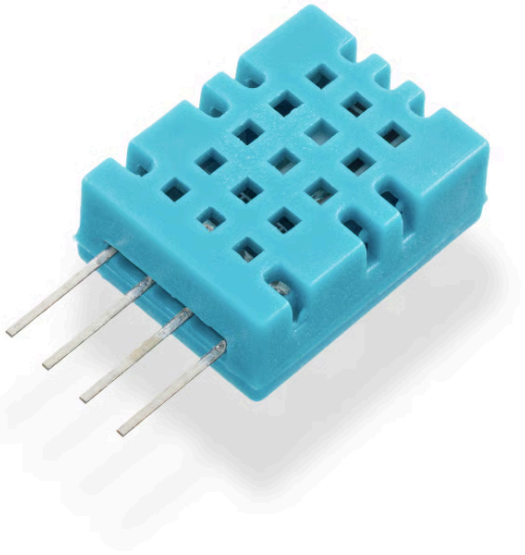
Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Bitmap encoding

What is a bitmap?

- A bitmap image is made up of squares called **pixels**
- A pixel is the **smallest element** of a bitmap image
- Each pixel is **stored as a binary code**
- Binary codes are **unique** to the **colour** in each pixel
- A typical example of a bitmap image is a photograph



Humidity sensor bitmap image

- The more colours and more detail in the image, the **higher the quality of the image** and the **more binary that needs to be stored**

Image vs screen resolution

- **Image resolution** is the **total amount of pixels** that make up a bitmap image
- The image resolution is calculated by **multiplying the height and width** of the image (in pixels)
- In general, the **higher the resolution the more detail in the image** (higher quality)
- **Screen resolution** refers to the **total amount of pixels horizontally in a display**, such as:
  - **Computer monitors** - 1440p means 1440 pixels horizontally compared to 4K which is 3840 pixels (roughly 4 thousand)

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- **TVs** - HD (high definition) televisions have a screen resolution of 1080p, 1080 pixels horizontally compared to newer 4K televisions with 3840 pixels
- **YouTube** - The quality button allows a user to change the video playback resolution from 144p (144 pixels horizontally) up to 4K

- Another consideration of screen resolution is the **physical size** of the of the display
- The **number of pixels per square inch** (**PPI**) is known as **pixel density**
- Pixel density can mean images to **need to be scaled up or down to fit**, sometimes causing loss in quality



Case Study

- A consumer purchases a new **65" 4k television**
- The screen resolution is **3840 x 2160**
- To calculate the pixel density of the screen we **add together the squares of the resolution**
  - $(3840^2 + 2160^2) = (14\,745\,600 + 4\,665\,600) =$ **19 411 200**
  - Find the **square root** ($\sqrt{19\,411\,200}$ = **4405.814**)
  - **Divide by the screen size** (4405.814 / 65 = **68**)
- The television has a **pixel density of 68** pixels per inch (**PPI**)
- Watching **4K content** from a normal viewing distance means the image will appear **crisp and sharp**
- However, if you sit **too close**, you may start to see:
  - The **pixel grid**
  - A **loss of fine detail**
- Modern smartphones have **very high screen resolutions** in **much smaller screens**
- This gives them a much **higher PPI**, often over **300 PPI**
- This means they can be viewed **up close without losing quality** or seeing pixelation

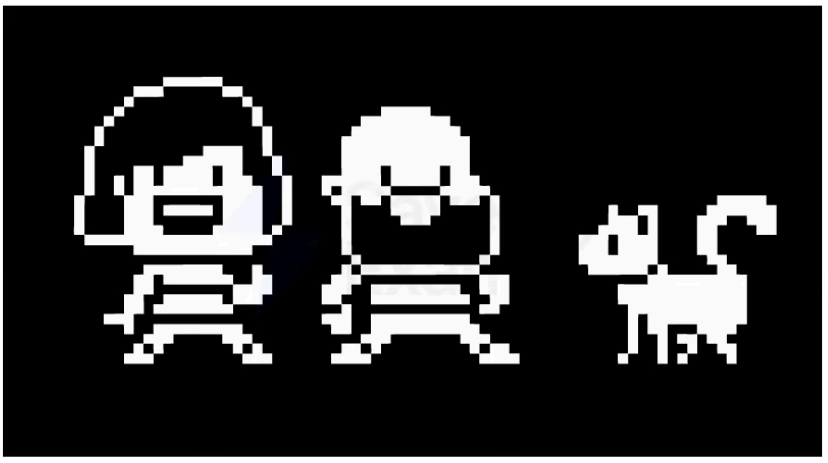
Colour/bit depth

- Colour depth is the **number of bits stored per pixel** in a bitmap image
- The colour depth is dependent on the **number of colours needed** in the image
- In general, the **higher the colour depth the more detail in the image** (higher quality)
- In a black & white image the **colour depth would be 1**, meaning 1 bit is enough to create a unique binary code for each colour in the image (1=white, 0=black)



Your notes

Study copy distributed by EduResource ZA | eduresourcza.com



Copyright © Save My Exams. All Rights Reserved

Your notes

- In an image with a **colour depth of 2**, you would have 00, 01, 10 & 11 available binary codes, so **4 colours**



Copyright © Save My Exams. All Rights Reserved

- As **colour depth increases**, so does the **amount of colours available** in an image
- **Colour depth** can also refer to the **number of colours** that can be represented in an image
- It is calculated using the formula:
  - **Colour depth = 2^n (where n = number of bits)**

Colour/bit depth	Number of colours	Example
1 bit	$2^1 = 1$	B&W
2 bit	$2^2 = 4$	Icons/logos
4 bit	$2^4 = 16$	Early computer graphics
8 bit	$2^8 = 256$	GIFs, retro games

Study copy distributed by EduResource ZA | eduresourcza.com

24 bit (True colour)	2^{24} = 16 777 216	High-quality images



Your notes

Calculating the size of a bitmap file

How do you calculate the size of a bitmap image?

- Estimating the size of a bitmap image can be carried out with the following formula:
  - **Resolution x colour/bit depth**

Example

Image Files		
(Resolution) **x** (Colour Depth)		
Size of bitmap image =		
Resolution	**250,000**	Resolution = width x height
Colour depth	**24 bits**	24 bits = 3 bytes
250,000 x 24	**=** **(bit to bytes) /8** **(bytes to KiB) /1024** **or** **(bytes to KB) /1000**	**6,000,000 bits** **750,000 bytes** **732 KiB** **or** **750 KB**
250000 x 3	= **(bytes to KiB) /1024** **or** **(bytes to KB) /1000**	**750,000 bytes** **732 KiB** **or** **750 KB**

- When bitmap images are saved, **a file header is created**
- This contains:
  - **File type (.bmp or .jpg)**
  - **File size**
  - **Image resolution**

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- **Colour depth**
- **Any type of compression if used**

Your notes

Worked Example

The following section of a bitmap image is 10 pixels wide and 5 pixels high. In this example, each colour is represented by a letter, e.g. O is orange.

The complete image can have up to 256 colours

B	B	B	B	B	B	B	B	B	B
Y	Y	P	Y	Y	Y	P	Y	Y	Y
R	R	M	R	P	K	T	T	R	R
B	O	P	Y	Y	Y	P	G	P	P
R	O	R	P	P	P	R	R	R	R

a) Identify the smallest number of bits that can be used to represent each colour in the complete bitmap image. **[1]**

b) Calculate an estimate for the file size of the section of the bitmap image shown, giving your answer in bytes. Use your answer from **part a** and show your working. **[2]**

Answer

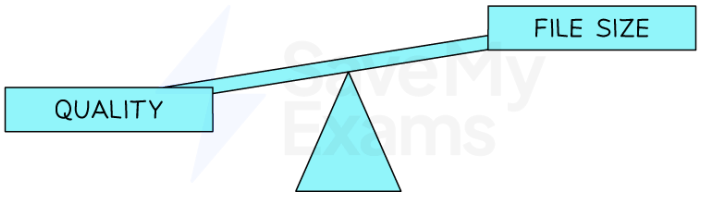
- 8 [1 mark]
- 10 x 5 = 50(pixels), 50 x 8(bits) = 400 // 400 ÷ 8(bits) = 50 [1 mark]
- 50 (bytes) [1 mark]

Image quality & file size

What is the relationship between image quality and file size?

- As the **resolution and/or colour depth increases**, the **bigger the size of the file** becomes on secondary storage
- The **higher the resolution**, the **more pixels** are in the image, the more bits are stored
- The **higher the colour depth**, the **more bits per pixel** are stored
- Striking a **balance between quality and file size** is always a consideration
- **Compression** can be used on high quality images to help reduce the file size

Study copy distributed by EduResource ZA | eduresourcza.com



Copyright © Save My Exams. All Rights Reserved

Your notes

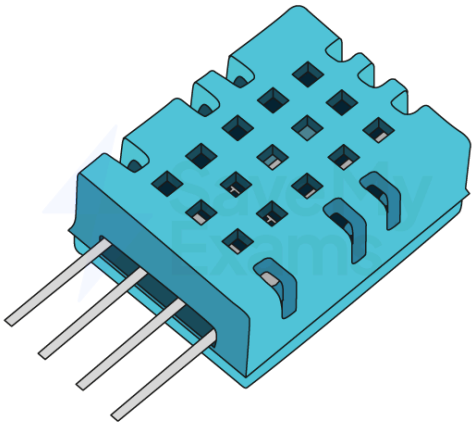
Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Vector graphics

What is a vector graphic?

- A vector graphic is created from **mathematical equations** and **points**
- Only the **mathematics used** to create the image are stored
- For example, to create a circle the data stored would be:
  - **Centre point (x, y coordinates)**
  - **Radius**
- Typical examples of vector images are **logos and clipart**



Copyright © Save My Exams. All Rights Reserved

- Vector images are infinitely **scalable**
- Ideal for situations where the same image **will be made bigger and smaller** and a **loss of quality is unacceptable**
  - For example, the same logo used on both a pencil and a billboard
- A vector graphic contains a **drawing list** (included in the file header)
- A typical drawing list will contain:
  - **Commands used to create each object in the image**
  - **Attributes that define the properties of each object**
  - **Relative position of each object**

Study copy distributed by EduResource ZA | eduresourcza.com

- Dimensions of the image **are not defined**, meaning scaling up does not result in a loss of quality



Your notes

Bitmap vs vector

Feature	Bitmap Images	Vector Images
Made from	Pixels (tiny coloured squares)	Shapes and lines using maths (vectors)
File size	Larger (depends on resolution)	Smaller (uses formulas, not pixels)
Scalability	Loses quality when resized (pixelates)	Can be resized without losing quality
Best for	Photographs, detailed images	Logos, icons, text graphics
Editable with	Paint software (e.g. Photoshop, GIMP)	Drawing software (e.g. Illustrator, Inkscape)
Common file types	.jpg, .png, .bmp, .gif	.svg, .ai, .eps

- When **choosing** to use a bitmap or vector based image, the following should be considered:
  - **Does the image need to be resized?**
  - **Does the image need to be drawn to scale?**
  - **Does the image need to look real?**

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

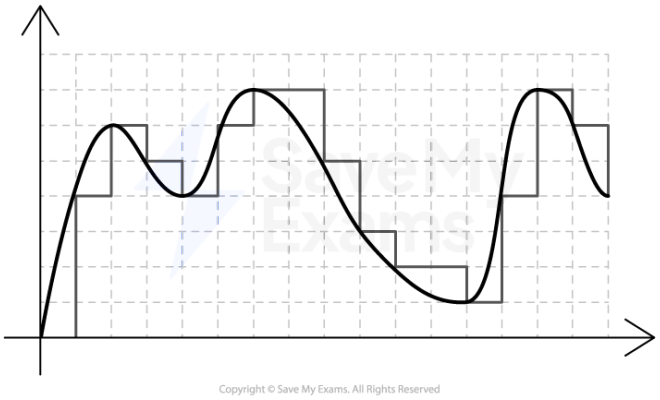
Sound encoding

How is sound encoded in a computer system?

- Computers represent all data in binary, including sound that we **record** using a microphone (input) or sound that we **playback** from a speaker (output)
- For this to happen, analogue sound must be **sampled and stored**

Analogue sound

- Sound waves begin as **analogue** and for a computer system to understand them they must be converted into a **digital form**
- Measurements of the original sound wave are **captured** and **stored as binary** on secondary storage
- This process is called **Analogue to Digital conversion** (**A2D**)
- The process begins by measuring the loudness (**amplitude**) of the analogue sound wave at a point in time, this is called **sampling**
- The higher the amplitude, the **louder** the sound
- Each measurement (**sample**) generates a value which can be represented in **binary** and **stored**
- Using the **samples**, a computer is able to create a **digital version** of the original analogue wave
- The digital wave is **stored** on **secondary storage** and can be played back at any time by reversing the process


Copyright © Save My Exams. All Rights Reserved

- In this example, the grey line **represents the digital wave** that has been created by taking samples of the original analogue wave

Study copy distributed by EduResource ZA | eduresourcza.com

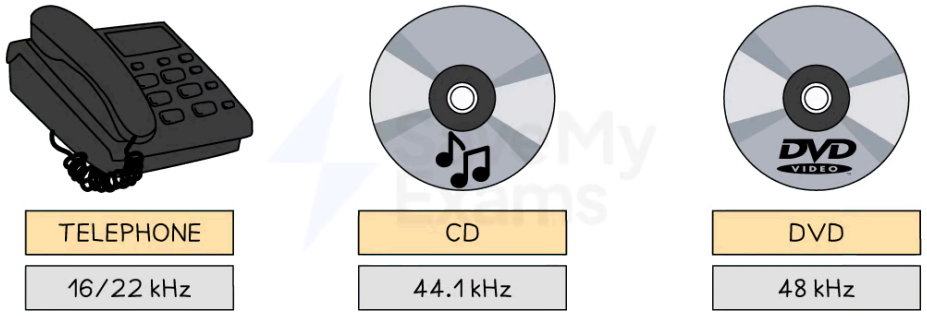
- In order for the digital wave to **look more like the analogue wave** (black line) the **sampling rate**, **sampling resolution** and **sample interval** can be changed

Your notes

Sampling rate vs sampling resolution

Sampling rate

- The sampling rate is the amount of **samples taken per second** of the analogue wave
- Samples are taken for the **duration** of the sound
- The sampling rate is measured in **Hertz** (**Hz**)
- **1 Hertz is equal to 1 sample** of the sound wave



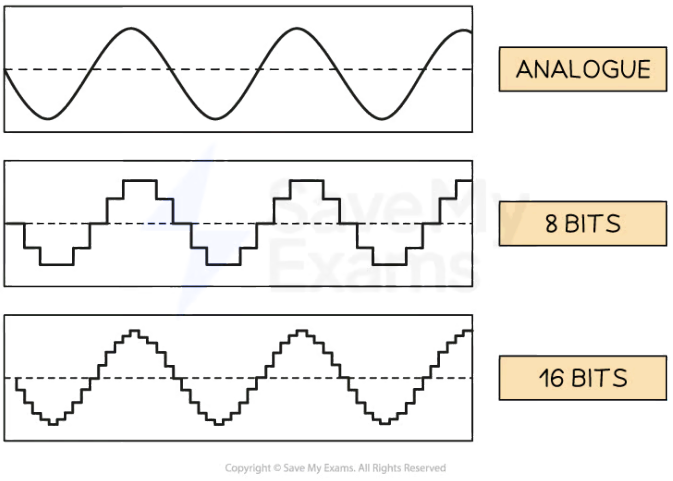
Copyright © Save My Exams. All Rights Reserved

- The sample rate of a typical audio CD is **44.1kHz** (44,100 Hertz or 44,100 samples per second), a **sampling resolution** of 16 bits and is recorded in **stereo sound**
- Using the graphic above to compare common sampling rates, the question, "*Why does telephone hold music sound so bad?*" can now be answered

Sampling resolution

- Sampling resolution is the **number of bits** used to represent **each sound sample**
- Sampling resolution is closely related to the **bit depth of a bitmap image**, they measure the same thing in different contexts

Study copy distributed by EduResource ZA | eduresourcza.com



Copyright © Save My Exams. All Rights Reserved

Your notes

- In the example above, the higher the sample resolution, the **closer to the original** sound wave the digital version looks

Impact of sampling settings

What are the impacts of sampling settings?

Factor	Effect of playback quality	Effect on file size
Sampling rate	↑**higher** = more detail, better sound quality	↑**higher** = more data, larger file size
Sampling resolution	↑**higher** = bigger range, better sound quality	↑**higher** = more data per sample, larger file size

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Compression

Contents

- Compression
- Compression techniques

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

The need for compression

What is compression?

- Compression is **reducing the the size of a file** so that it **takes up less space on secondary storage**
- There are scenarios where compression may be needed, such as:
  - **Maximise the amount of data you can store** on a digital device such as a mobile phone or tablet
  - **Minimise the transfer time** of data being uploaded, downloaded or streamed across a network such as the Internet
- Compression can be achieved using two methods, **lossy** and **lossless**

Lossy vs lossless

What is lossy compression?

- Lossy compression is when **data is lost** in order to **reduce the size** on secondary storage
- Lossy compression is **irreversible**
- Lossy can **greatly reduce the size** of a file but at the **expense of losing quality**
- Lossy is only suitable for data **where reducing quality is acceptable**, for example images, video and sound
- In photographs, lossy compression **will try to group similar colours together**, reducing the amount of colours in the image without compromising the overall quality of the image

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

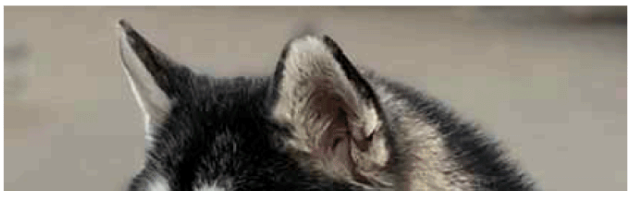
Your notes



ORIGINAL JPG
824KB



50% LOSSY COMPRESSION
76KB



Study copy distributed by EduResource ZA | eduresourcza.com



80% LOSSY COMPRESSION
38KB

Copyright © Save My Exams. All Rights Reserved

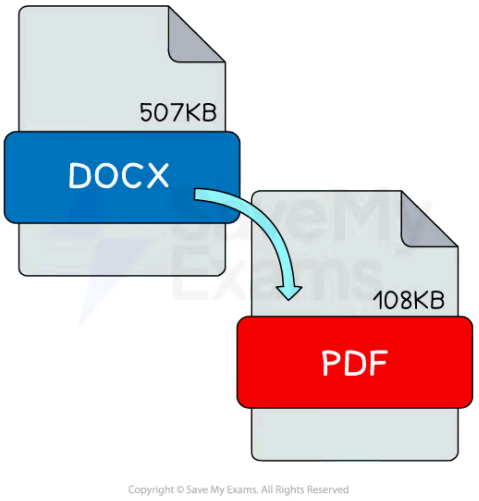
- In the images above, lossy compression is applied to a photograph and **dramatically reduces the file size**
- **Data has been removed** and the **overall quality has been reduced**, however it is acceptable as it is difficult to visually see a difference
- Lossy compressed photographs **take up less storage space** which means you can store more and they are quicker to share across a network

What is lossless compression?

- Lossless compression is when data is **encoded** in order to **reduce the size on secondary storage**
- Lossless compression is **reversible**, the file **can be returned to its original state**
- Lossless can reduce the size of a file but not as dramatically as lossy
- Lossless can be used on all data but is more suitable for data where **a loss in quality is unacceptable**, for example documents
- In a document, lossless compression uses algorithms to analyse the contents looking for **patterns** and **repetition**

Your notes

Study copy distributed by EduResource ZA | eduresourcza.com



Copyright © Save My Exams. All Rights Reserved

Your notes

- In the image above, lossless compression is **automatically applied** to document formats such as **DOCX** and **PDF** with a different rate of success
- When you open a lossless compressed document the **decompression process reverses the algorithms** and returns the data back to its **original state**
- Lossless compressed documents **take up less storage space** which means you can store more and they are quicker to share across a network

Study copy distributed by EduResource ZA | eduresourcza.com



Your notes

File compression

MPEG-3 (MP3) – Audio compression

- MP3 uses **audio compression** to reduce file size
- A typical music file can be reduced by **up to 90%** (e.g. 80 MB to 8 MB)
- MP3 files are used on:
  - **MP3 players**
  - **Computers**
  - **Smart phones**
- Music can be:
  - **Downloaded or streamed** from the internet
  - **Converted from a CD** to MP3 format
- MP3 quality isn't as high as the original CD, but it's **good enough for most people**
- MP3 uses **perceptual music shaping** to remove sounds we don't notice:
  - Frequencies **outside human hearing range**
  - **Quieter sounds** that are masked by louder ones
- This makes the file smaller **without a big drop in quality**
- MP3 is a **lossy** format:
  - Some original data is **permanently lost**
  - You **can't get the original file back**

MPEG-4 (MP4) – Multimedia file format

- MP4 is similar to MP3, but stores **more than just audio**
- It can store:
  - **Music**
  - **Videos**
  - **Photos**
  - **Animations**
- Commonly used for **streaming videos** over the internet
- MP4 uses **compression** to keep file size small

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- Maintains **high quality** without noticeable loss to the viewer



Your notes

Images

- **Bitmap images can be compressed**, the file size and quality are reduced
- **JPEG** is the most common file format for bitmap images
- The JPEG **compression algorithm creates a new file**, so the original file can no longer be used
- **Vector** graphics can also be compressed but **with limited success**
- **Scalable vector graphics** (.svg) use an XML text file which means that allows them to be compressed.

Run-length encoding (RLE)

What is run-length encoding?

- Run-length encoding (**RLE**) is a form of data compression that **condenses identical elements into a single value** with a count

Text data

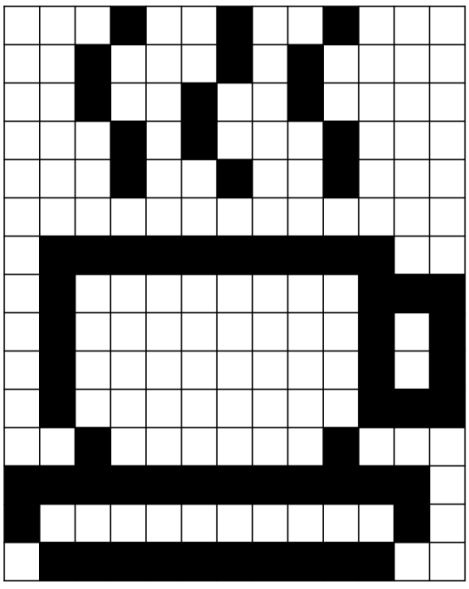
- For text data containing the string "**AAAABBBCCDAA**", the plain RLE encoding would be "**4A3B2C1D2A**"
- The string has:
  - four 'A's (**4A**)
  - three 'B's (**3B**)
  - two 'C's (**2C**)
  - one 'D' (**1D**)
  - two 'A's (**2A**)
- To represent this in binary, the **count is stored in a fixed size binary format** (e.g. 7 or 8 bits)
- The character is stored using its **ASCII** value (7 bits)
- The binary RLE representation of **4A** would be **0000100 1000001**
  - **0000100** - binary for the count (4)
  - **1000001** - binary for 'A' (65)

Images

- Run-length encoding (**RLE**) uses **frequency/data pairs** to compress **bitmap image** data
- For example, the following bitmap image with a **bit depth of 1 bit** would have the following binary bit pattern

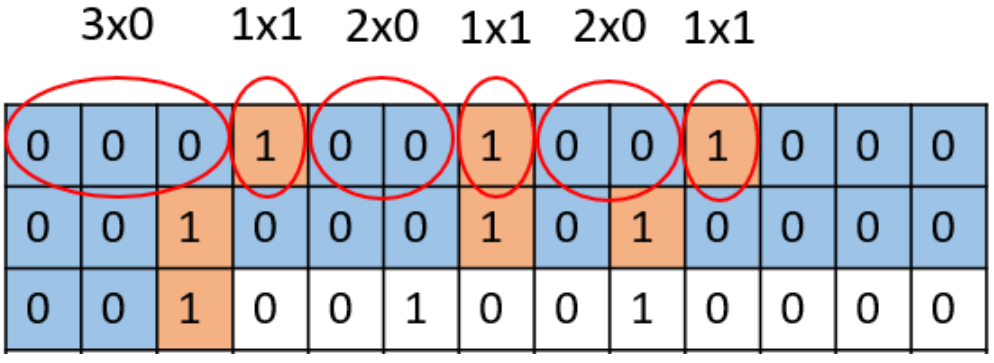
© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Bitmap	Bit pattern
	0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 0 0

Your notes

- Using RLE we **group pixel colours** and can create frequency/data pairs as follows
  - 30, 11, 20, 11, 20, 11, 50, 11, 30, 11, 10, 11, 60, 11.....



- **3 x 0 =** 3 x white, **1 x 1 =** 1 x black
- Data pairs **can carry over on to the next line**, e.g. end of first line and start of second line is 5 x 0 (**5 x white**)
- The **original** file would need to use **195 bits** of storage space
- The **RLE compressed** file would need use **38 bits** of storage space, **a saving of 157 bits (80%)**
- For images using more **than two colours**, the RGB value for each colour is stored along with the count
- For example, an image using the following four colours:

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Colour	Red	Green	Blue
Black	0	0	0
White	255	255	255
Red	255	0	0
Green	0	255	0



Your notes

- The data stored could look like:
  - 10 0 0 0 5 255 0 0 3 0 255 0
- **Ten black pixels** ((10) 0 0 0) - count + RGB
- **Five red pixels** ((5) 255 0 0) - count + RGB
- **Three green pixels** ((3) 0 255 0) - count + RGB

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Networks & the Internet

Contents

- Network types
- Network models
- Network topologies
- Cloud computing
- Transmission media
- Network hardware
- Ethernet
- Bit streaming
- The Internet
- IP addressing
- Locating resources on the World Wide Web

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Purpose & benefits of networking

What is a network?

- A network is two or more **interconnected devices** (such as computers, printers, and servers) designed to:
  - **share resources**
  - **exchange data**
  - **communicate with each other**
- The purpose of a network is to enable **data** and **resource** sharing, **communication**, and **collaboration**

Advantages and disadvantages of networks

Advantages	Disadvantages
Devices (e.g. printers) can be **shared**, reducing costs	**Expensive setup** – cabling and servers can cost a lot
Software licences for networks are cheaper than for individual computers	**Difficult to manage** – large networks need skilled administration
Users can **share files and data** easily	If the **server fails**, the whole network can be affected
Access to **reliable, central data** (e.g. from a file server)	**Malware or hacking** can affect the entire network
Files can be **backed up centrally** each day	**Security risks** increase if connected to a larger WAN
Users can **communicate** via email or messaging	
A **network manager** can control access rights and internet usage	

LANs & WANs

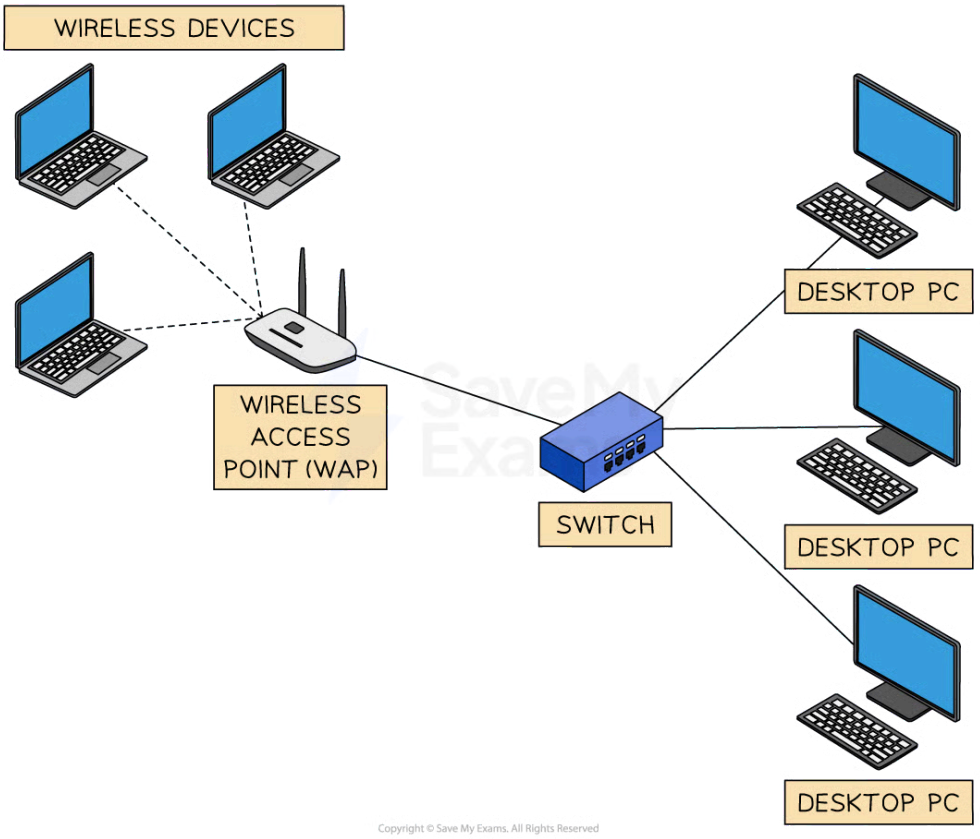
What is a local area network (LAN)?

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- A local area network (LAN) is a network which has a **small geographical area** (under 1 mile)
- All of the hardware is **owned by the company/organisation/household using it**
- LANs will use unshielded twisted pair (**UTP**) cable, fibre optic cable or wireless connections (**Wi-Fi**)

Your notes



Advantages and disadvantages of LANs

Advantages	Disadvantages
Allows **centralised** management of updates, backups and software installations	If **hardware fails**, the network may not function properly or even at all
Can **secure** its devices with the use of firewalls, antivirus software and other security features to prevent unauthorised access	Networks are more **prone to attacks** than standalone computers
Allows users on the network to **share resources** such as printers and other peripherals	Access to data and peripherals can be slow depending on **network traffic**

Study copy distributed by EduResource ZA | eduresourcza.com

Allows the users of the network to **collaborate** and **share files and folders**	Require **maintenance** to ensure that software is up to date, upgrades and backups which can be costly



Your notes

Wireless Local Area Network (WLAN)

- A wireless local area network is a local area network where devices connect to the network **wirelessly** instead of using cables
- Extra hardware, **wireless access points** (**WAPs**) or **hotspots** are connected to the network so that users can connect using Wi-Fi

Advantages and disadvantages of WLANs

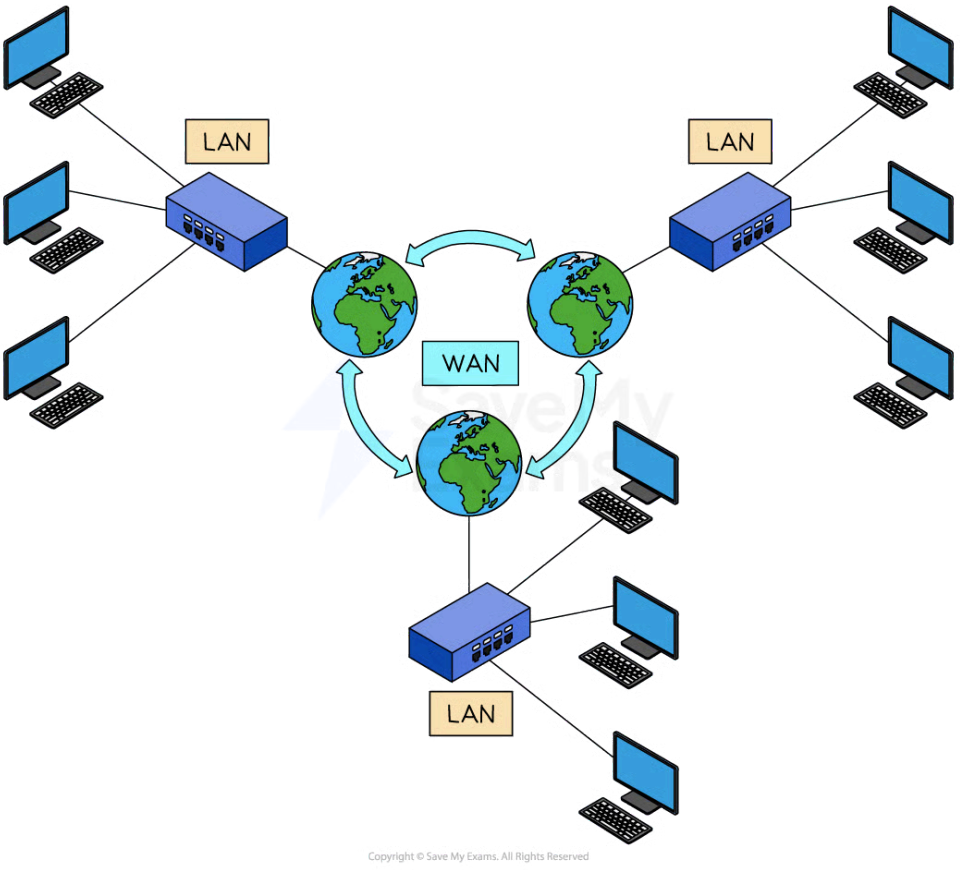
Advantages	Disadvantages
Allows users to connect anywhere that is in the range of a **Wireless Access Point (WAP)** without the need for additional hardware or wiring.	**Limited** in their coverage and can be further affected by walls and other structures
Can be used in a **variety of environments** both indoors and out making them highly flexible	**Bandwidth** speeds can become an issue in high traffic areas
Additional wireless access points can be added relatively easily resulting in additional users being able to use the network or **increased network coverage**	**Interference** from other devices which can affect performance and connectivity
Access to **peripherals** such as printers	Vulnerable to **security threats** due to wireless signals being intercepted

Wide Area Network (WAN)

- A wide area network (WAN) is a network which has a **large geographical area** (over 1 mile)
- They are a **collection of LANs joined together**
- The computers on a WAN are **connected via routers**
- The hardware used to connect the networks together is **not all owned by the company/organisation/household using it**.
- For example, telephone lines **owned by telecommunication companies**
- WANs will use **fibre optic** cable, **telephone lines** and **satellite** to connect the LANs together

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com



Copyright © Save My Exams. All Rights Reserved

Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Client-server & peer-to-peer

What is a client-server network?

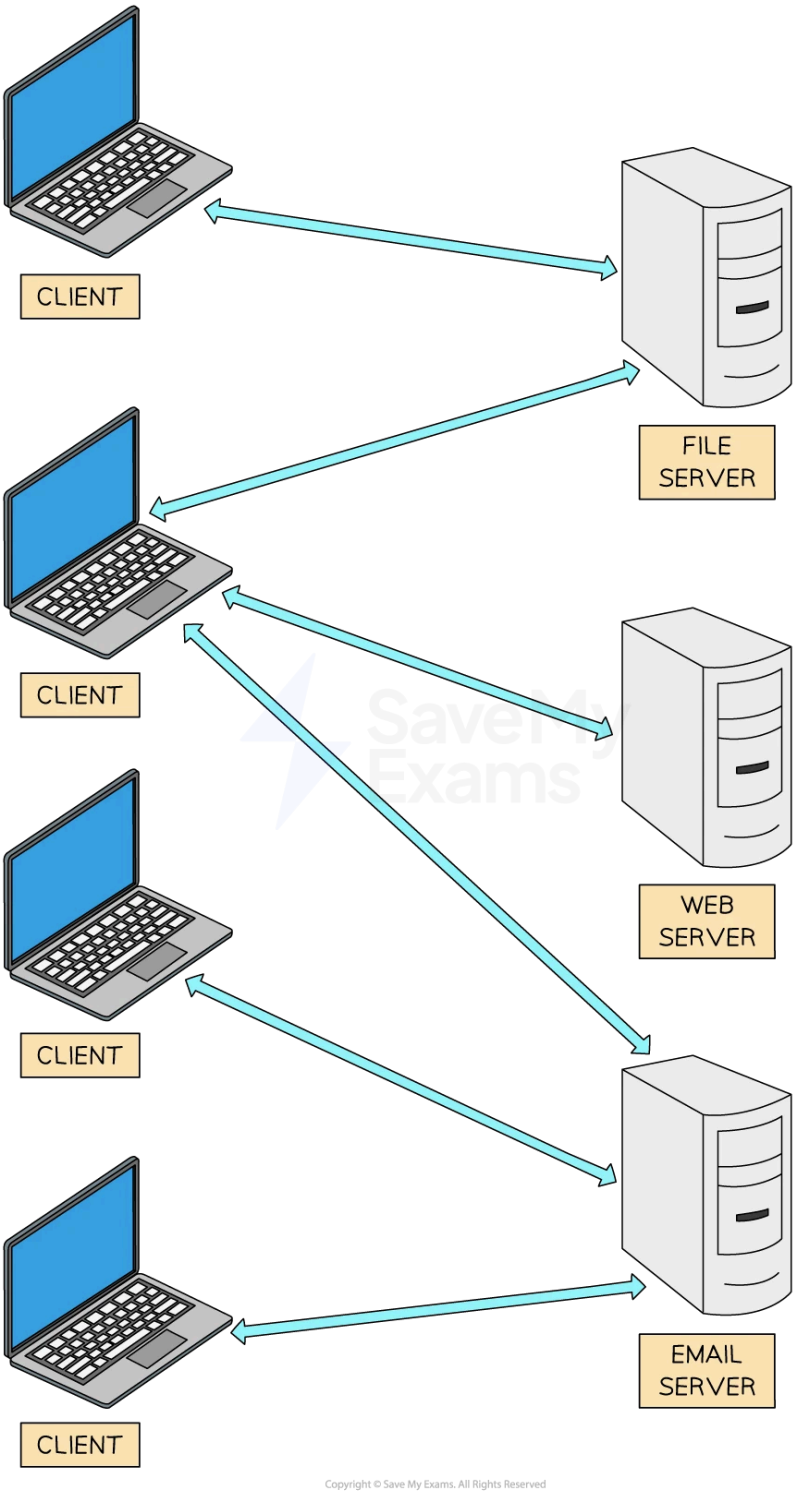
- A client-server network is when **powerful and reliable 'server' computers** control the network and **'serve' the clients** with services such as files, email, web access, etc
- Clients **connect to the servers** to access network services
- In this setup, the server **hosts**, **delivers** and **manages** most of the resources and services to be consumed by the clients

Benefits	Drawbacks
Easier **central management**	**Single point of failure** - if the server goes down, services could be unavailable
Scalability: new clients can be added easily	It can be **expensive** to set up and maintain - often need dedicated teams of people to maintain them
Higher reliability as resources are managed centrally	

- A client-server network is typically used by **larger organisations** where **centralised control is needed**, and reliability and security are paramount

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com



Client computers connected to different servers

Your notes

When to use a client-server network

- The choice between client-server and peer-to-peer depends on the **specific needs and resources** of the network in question
- **Security**, **cost**, **ease of setup**, and **maintenance** requirements should be considered

Study copy distributed by EduResource ZA | eduresourcza.com

Peer-to-peer



Your notes

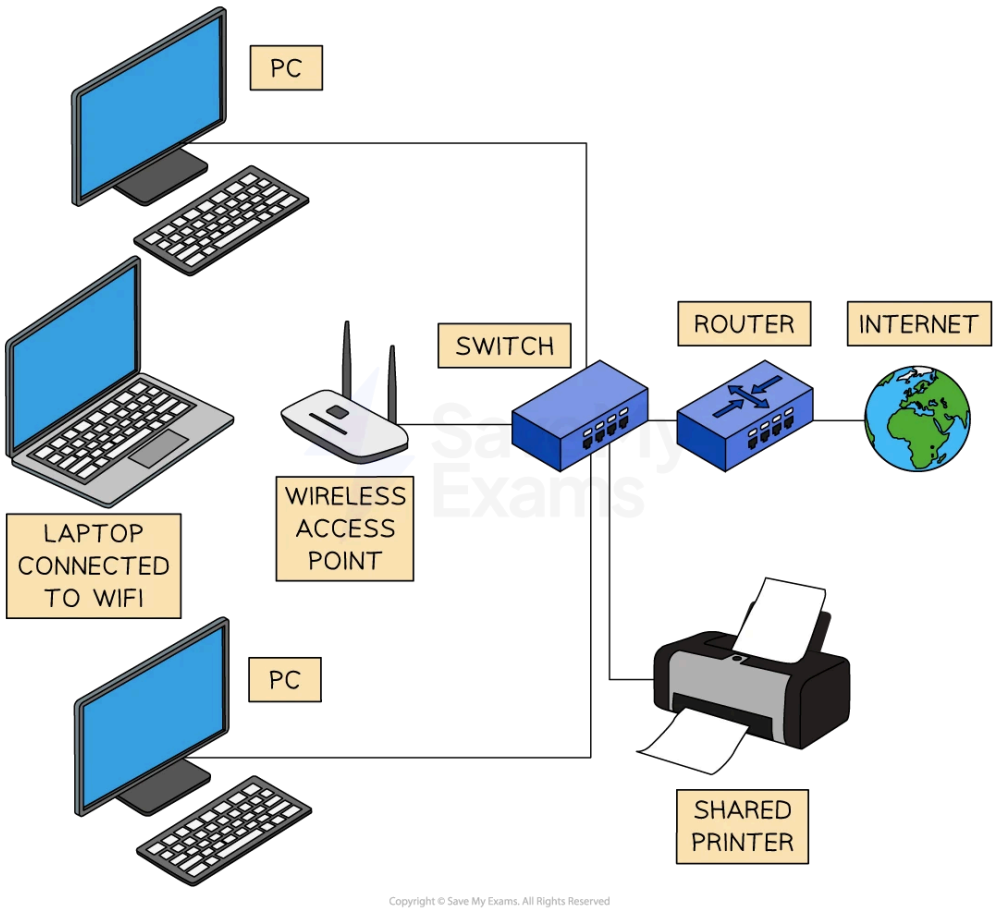
- This is the **simplest** type of network
- In this setup, all **computers in the network share equal responsibility,** and there is no central server
- All **machines have equal status**
- Each machine is the **responsibility of that machine's user** in terms of security, backup, etc.
- **Data is often spread around the network,** with each user being responsible for their data

Benefits	Drawbacks
Easy to set up and less expensive than client-server as no administrative staff are needed	**Lack of central control** can lead to security issues and vulnerabilities
No dependency on a central server	**Not suitable for large networks** as it can have performance issues
Data can be shared directly between systems without the need for a central server	

- A peer-to-peer network is typically used in **home networks**, by **small businesses**, or for **specific applications** like file sharing

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com



Peer to peer network example setup

Your notes

Worked Example

The Internet uses a client-server model.

Describe the role of clients and servers on the Internet **[2]**

Answer

- Web pages/files are saved on servers [1 mark]
- Clients send requests to the web servers [1 mark]
- Web servers process the requests [1 mark]
    - ...and return the results to the client [1 mark]
    - the client displays the results to the user [1 mark]

Thin-client & thick-client

What is a thin-client?

- A **thin client** relies on a **permanent connection** to a server to function

Study copy distributed by EduResource ZA | eduresourcza.com

- Can be **hardware** (a device) or **software** (an app or program)
- Depends on the **processing power** of another computer or server
- **Cannot function** without being connected to a networked computer or server
- The server can be part of a **LAN** (local area network) or a **WAN** (wide area network)
- Examples of thin-clients include:
  - **Cloud-based apps** like Google Docs or Microsoft 365
  - **Remote desktop software** (e.g. Chrome Remote Desktop, Citrix)
  - **Supermarket POS systems** connected to a central server for prices, stock, and payments



Your notes

Thick-client

- Works **independently** without needing constant connection to a server
- Can be a **hardware device** or **software** installed on a local machine
- Uses the **processing power of its own device** to run applications
- May connect to a network for updates or data sharing, but **can function offline**
- Often used where **performance, speed or offline access** is important
- Examples of thick-clients include:
  - **Installed software** like Adobe Photoshop or Microsoft Word
  - **Standalone games** installed and played on a PC or console
  - **School or office PCs** running full applications locally
  - **Laptops** that can run applications without an internet connection

Feature	Thin-client	Thick-client
Dependence on server	Needs a **permanent connection** to a server to function	Can work **independently** of a server
Processing power	Uses the **server's processing power**	Uses its own **local processing power**
Functionality when offline	**Does not work** without a connection	Can **function offline**
Examples	Google Docs, Microsoft 365, Remote Desktop, supermarket POS systems	Microsoft Word, Adobe Photoshop, standalone games, school laptops

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Best for	**Centralised control**, low-cost devices, shared environments	**Performance**, flexibility, offline access
Network type	Part of a **LAN** or **WAN**	May connect to a network, but **not reliant** on it



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Network topologies

Your notes

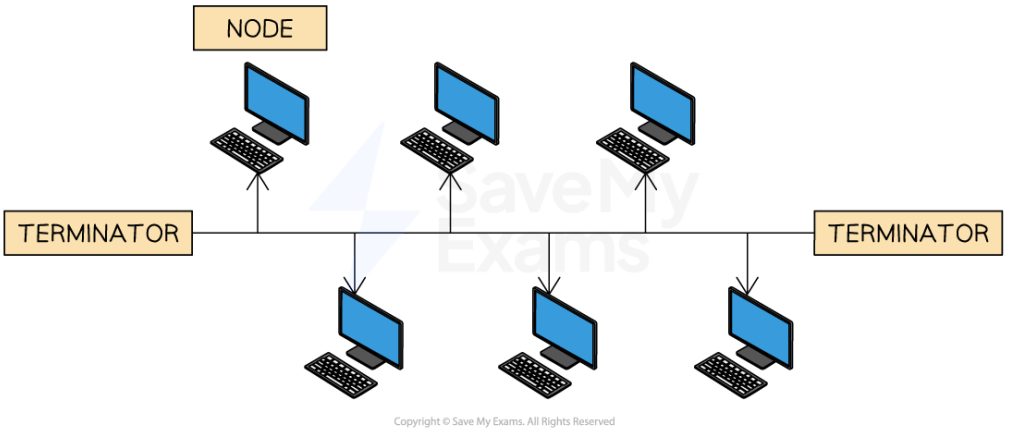
What is a network topology?

- A network topology is the **physical structure of the network**
- It defines **how the network hardware will be arranged** to create the network
- Topologies to understand for the exam are:
  - **Star topology**
  - **Bus topology**
  - **Mesh topology**
  - **Hybrid topology**

Bus

What is a bus topology?

- A bus topology is all **devices connected by one single 'bus' cable**, terminated at each end
- The terminators **stop the signal bouncing back** and **causing errors**
- A bus topology works by each device:
  - **'Listening'** to electrical signals
  - **Checking** data packets for their specific address
  - **Ignoring** data packets it does not recognise
- A bus topology has been replaced by much more **efficient network topologies** such as the **star topology**



Copyright © Save My Exams. All Rights Reserved

Bus topology – advantages vs disadvantages

Study copy distributed by EduResource ZA | eduresourcza.com

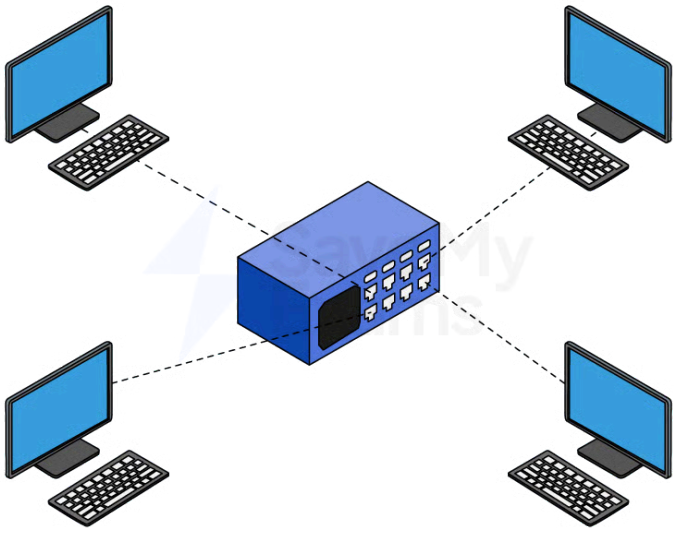
Advantages	Disadvantages
Easy and cheap to set up – only one cable is needed	**Low security** – data is visible to all devices
Does not rely on other network hardware (e.g. central server or switch)	**Slow data transfer** – prone to data collisions
	If the cable breaks, the **whole network is affected** (central point of failure)

Your notes

Star

What is a star topology?

- A star topology **has a central switch** which all other devices connect to
- A switch is an **intelligent device** which **ensures that traffic only goes to the intended device**
- A star topology is commonly seen in most **homes**, **businesses**, **organisations** and **schools**


Copyright © Save My Exams. All Rights Reserved

- Traffic (packets) on a star topology is **handled differently depending on the device** used as the central node
  - A **hub** will send traffic to all devices on the network, if the address in the packet matches a device it will be accepted, otherwise it is ignored
  - A **switch** will only send traffic to devices on the network where the address in the packet is a match

Study copy distributed by EduResource ZA | eduresourcza.com

Star topology – advantages vs disadvantages

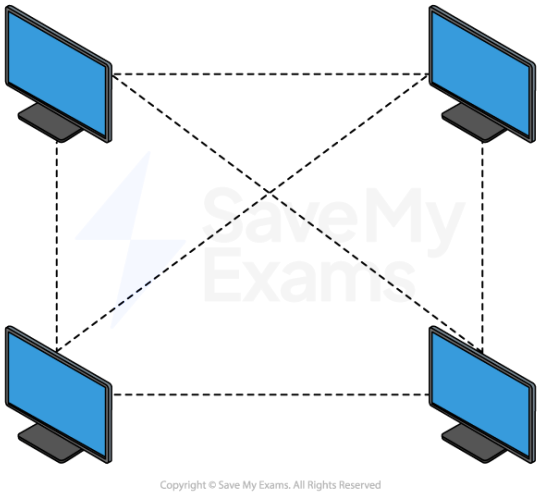
Advantages	Disadvantages
If **one cable fails**, the rest of the network still works	The **central switch is a single point of failure**
A problem with **one device** (e.g. a school computer) won't affect others	If the **switch stops working**, **all devices lose connection** to the network and its resources
More **reliable** than other topologies like bus or ring	Can be **expensive** to install due to the switch and extra cabling (not mentioned but useful add-on)

Your notes

Mesh

What is a mesh topology?

- A mesh topology allows **all computers to be connected to all other computers**
- This is known as a **full mesh** network topology
- LANs can make use of mesh networks however, they are **more commonly seen in** IoT devices such as **wearable technology** and **smart home** devices


Copyright © Save My Exams. All Rights Reserved

- Traffic (packets) on a mesh topology can follow two different methods:
  - **Routing** - devices are given routing logic (act like a router) to ensure packets are sent to the correct device in the shortest route
  - **Flooding** - all packets are sent to all devices using no routing logic, this can lead to network flooding causing performance issues

Study copy distributed by EduResource ZA | eduresourcza.com

Mesh topology – advantages vs disadvantages

Advantages	Disadvantages
If **one cable fails**, the network still works – data can use an **alternate route**	Requires **lots of hardware, cables, and switches**
Provides **high fault tolerance** – multiple paths for data to travel	**High cost** to set up due to the amount of equipment needed
Very **reliable** – good for critical systems	**Hard to scale** – adding new devices is more complex than in a star topology
	A **partial mesh topology** is often used as a more practical and cost-effective alternative



Your notes

Hybrid

What is a hybrid topology?

- A hybrid topology is a **mix of two of more different** network topologies
- For example, bus & star, star & mesh etc
- A typical use of a hybrid topology would be when there is a **need to join different networks** together
- Imagine a large education trust takes over three schools: **School A**, **School B**, and **School C**.
  - **School A** uses a **bus topology** to connect classroom computers with a simple shared cable
  - **School B** uses a **star topology** to link all devices to a central switch in the IT room
  - **School C** uses a **mesh topology** to make sure every building (e.g. science block, library) stays connected, even if one cable fails
- When the trust wants to connect all three schools into a single network, it uses a **hybrid topology**
- This allows:
  - Each school to **keep its own existing setup** (bus, star, or mesh)
  - The schools to be **linked together efficiently**
  - **Easy expansion**, such as adding a new school or admin office without changing everything
  - The network to be more **flexible and reliable**, using the strengths of each topology

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Hybrid topology – advantages vs disadvantages

Advantages	Disadvantages
Combines the strengths of multiple topologies (e.g. reliability of mesh, simplicity of bus)	**Complex to design** – needs careful planning to combine different topologies
Flexible – different parts of the network can be optimised for different needs	**Expensive** – more hardware and maintenance may be needed
Scalable – new segments (e.g. a new site or building) can be added without changing the whole setup	**Troubleshooting can be harder** – different parts use different systems and rules
Reliable – failures in one part don't necessarily affect the rest of the network	



Your notes

© 2026 Save My Exams, Ltd.

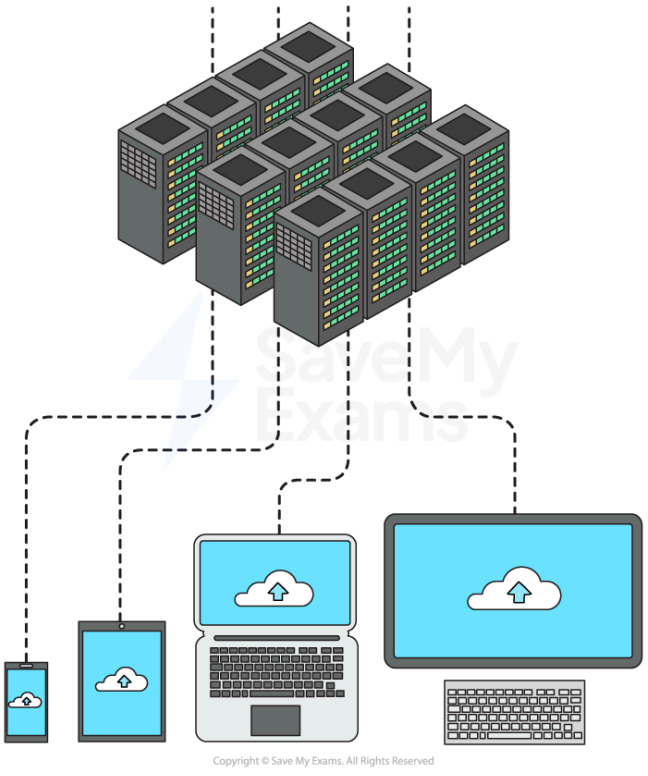
Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Cloud computing

What is cloud computing?

- Cloud computing is when **software, services or files are hosted entirely on remote servers**
- Accessed **through the internet**
- The two most common cloud computing examples are:
  - **Cloud storage**
  - **Cloud software**



Cloud storage

- Cloud storage is **long-term storage of data** that resides in a **remote location**
- Accessible only via the **Internet** (WAN)
- Data is stored on **remote servers** in data centres
- Storage is usually on **HDDs** (magnetic) but increasingly **SSDs** (solid state)
- Examples include: Google Drive, Dropbox, OneDrive

Study copy distributed by EduResource ZA | eduresourcza.com

Cloud software



Your notes

- Cloud software is **hosted and managed remotely**
- The user **accesses the software online** (on demand)
- The provider handles:
  - **Maintenance**
  - **Upgrades**
  - **Security**
- Typically paid for via a **monthly fee** or **yearly subscription**
- Examples include: Google Docs, Microsoft 365, Adobe Creative Cloud



Worked Example

A company uses cloud computing.

Define cloud computing. **[1]**

Answer

- Accessing a service/files/software on a remote server [1 mark]

Benefits and drawbacks of cloud computing

Cloud storage

Benefits	Drawbacks
Accessibility – access files from anywhere with an internet connection	**Needs a reliable internet connection** – slow or no connection can stop access to files
Easy to **share and collaborate** with others	**Can be expensive** – especially for large amounts of data or long-term use
Works on **any device** with internet access	**Ongoing costs** – often requires a monthly or annual subscription
Scalability – storage can be increased or decreased as needed	May need to pay for **extra data transfer** (upload/download limits)

Study copy distributed by EduResource ZA | eduresourcza.com

Reliability – data is backed up across multiple servers	**Security risks** – data sent over the internet could be intercepted
Security features – encryption, multi-factor authentication	**Less control** – provider manages security, but you're still responsible for protecting user data
No need to **buy expensive storage hardware**	
No need to **hire specialist IT staff** – support is handled by the provider	
Eco-friendly – centralised data centres are more efficient than millions of local servers	



Your notes

Cloud software

Benefits	Drawbacks
Accessible anywhere – use the software on any device with an internet connection	**Needs internet access** – won't work properly offline
No installation needed – runs through a browser or app	**Performance depends on internet speed** – slow connections can affect usability
Automatic updates – provider handles all software updates and patches	**Ongoing costs** – usually requires a monthly or yearly subscription
Maintenance is handled – no need for specialist IT staff to manage or fix the software	**Data privacy concerns** – your data is stored on the provider's servers
Scalable – easy to upgrade or downgrade your software plan	**Less control** – provider decides when updates or changes are made
Security included – providers manage security features like firewalls and encryption	**Legal responsibility** – you're still responsible for how personal data is stored, even if it's hosted
Often works **across multiple devices** – laptops, tablets, phones	Some features may be **limited** compared to full versions installed on a local device

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Public & private clouds



Your notes

What is the difference between a public and a private cloud?

Type	Public Cloud	Private Cloud
Ownership	Servers are owned and managed by a **third-party provider**	Servers are owned and managed **by the company itself**
Access	Services are shared with **other organisations or users**	Access is **restricted** to that company or organisation
Examples	Google Drive, Microsoft 365, Dropbox	Company intranet, internal company cloud servers
Cost	**Cheaper** – shared infrastructure reduces cost	**More expensive** – company pays for hardware, maintenance, and staff
Control	**Less control** over data and security settings	**More control** – company manages its own data and security
Security	Relies on **provider's security measures**	Can apply **custom security policies** and have full oversight



Worked Example

Give **two** benefits and **one** drawback of using cloud computing. **[3]**

Answer

1 mark each from:

Public

- Computing services offered by 3rd party provider over the public Internet [1 mark]
- Public is open/available to anyone with the appropriate equipment/software/credentials [1 mark]

Private

- Computing services offered either over the Internet or a private internal network [1 mark]
- Only available to select users not the general public [1 mark]

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- Private is a dedicated/bespoke system only accessible for/from the organisation [1 mark]



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Wired vs wireless

What is a wired network?

- A wired network is a network where **physical cables** are used to join devices together and **transmit data**
- Computers can be connected to networks using many different types of wires to transmit data
- The most common types of cables in a wired network are:
  - **Ethernet**
  - **Fibre optic**
  - **Copper**

Advantages & disadvantages of wired networks

Advantages	Disadvantages
Speed - Fast data transfer	**Portability** - Can't move easily, location is limited by physical cable
Security - Better physical security	**Cost** - Need more cables to add a new device
Range - High (up to 100m), less susceptible to interference	**Safety** - Cables can be trip hazards, need routing along walls, under floors

What is a wireless network?

- A wireless network is a network where connections are made **using radio waves** to transmit data **through the air**
- The most common types of wireless connections are:
  - **Wi-Fi**
  - **Bluetooth**

What is Wi-Fi?

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes



- Wireless fidelity (**Wi-Fi**) is a common **standard** for wireless networks
- Wi-Fi is common in most homes and offices to connect devices such as **laptops**, **tablets** & **smart phones**
- Using Wi-Fi, devices communicate with a wireless access point (**WAP**), which can be a standalone device or built into a **router** or **switch**

What is Bluetooth?

- Bluetooth is another common **standard** for wireless networks
- Bluetooth is common in most homes and offices to connect devices such as **headphones**, **controllers**, **keyboards** & **mice**
- Bluetooth is used typically for a **direct connection** between **two devices**

Advantages and disadvantages of wireless networks

Advantages	Disadvantages
Portability - Easy to move around, location is only limited by range	**Speed** - Slower data transfer than wired

Study copy distributed by EduResource ZA | eduresourcza.com

Cost - Less expensive to setup and add new devices	**Security** - Less secure than wired
Compatibility - Most devices are manufactured with a built in wireless adapter	**Range** - Relies on signal strength to the WAP, signals can be obstructed (up to 90m)

Your notes

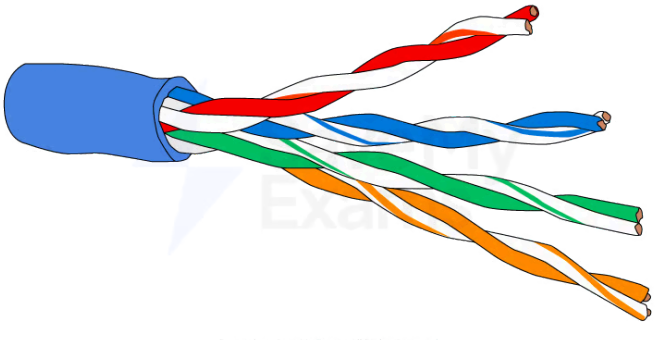
Types of media

What is transmission media?

- Transmission media is the **type of cable** used in wired connections
- Wired connections offer a **higher bandwidth** than wireless connections
- The main options for transmission media are:
  - **Twisted pair**
  - **Coaxial**
  - **Fibre-optic**

Twisted pair

- Carries electrical signals between devices on a local area network (**LAN**)
- Common in **offices and homes** to connect devices such a **desktop computers** & **servers**
- Allow **duplex** communication
- Slower **transfer rate** compared to coaxial and fibre-optic
- Can suffer from **external interference** (electromagnetic radiation)
- **Low cost**

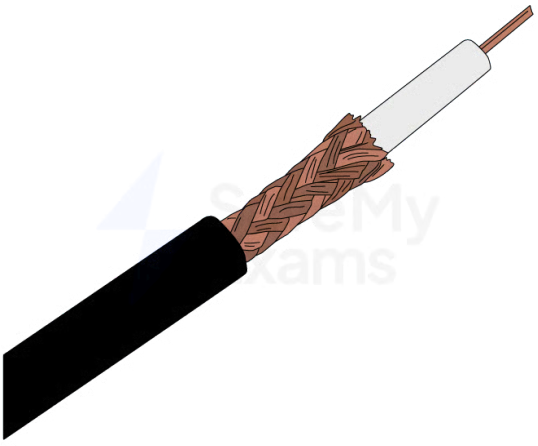


Copyright © Save My Exams. All Rights Reserved

Coaxial

Study copy distributed by EduResource ZA | eduresourcza.com

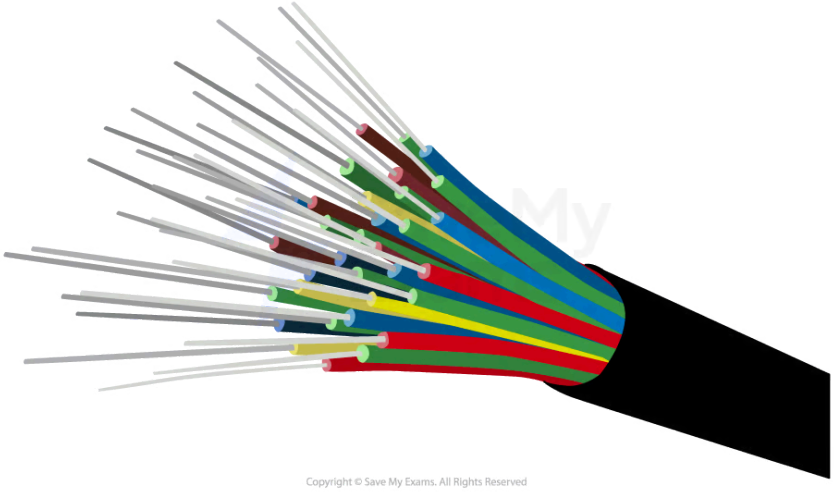
- **Used in telecommunication to transmit voice signals**, forming the traditional landline phone network
- Adapted to deliver **network traffic** on a wide area network (**WAN**) making the **internet possible**
- **Degrades over time** which **limits their range** compared to fibre optic
- **Suffers from interference** which can **disrupt data quality**
- Transmits data at a much **slower rate**, and has a much **lower bandwidth** compared to fibre optic



Copyright © Save My Exams. All Rights Reserved

Fibre optic

- Uses **light to transmit data** on a wide area network (**WAN**)
- Transmits data at **high speed** and has a **higher bandwidth** compared to copper cables
- **Does not suffer from interference** which makes them the **most secure option** to send **sensitive data**
- Can **cover a long distance** without any **degradation**, they can span cities and countries



Copyright © Save My Exams. All Rights Reserved

Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

Quick comparison

Feature	Twisted pair	Coaxial cable	Fibre optic
Used for	LAN connections in homes/offices	Voice signals (telecoms), early WAN/internet	High-speed WANs, internet infrastructure
Data transfer method	Electrical signals	Electrical signals	Light signals
Speed	Slower than coaxial and fibre	Slower than fibre	**Fastest**
Bandwidth	Low to medium	**Lower than fibre**	**Highest**
Signal degradation	Moderate	Degrades over time	**Minimal degradation** – works over long distances
Interference	Prone to electromagnetic interference	Also suffers from interference	**No interference** – more secure
Communication	Allows **duplex communication**	Supports duplex (but lower quality)	Full duplex
Cost	**Cheap**	Medium cost	**Expensive**
Typical use	Desktop PCs, printers, servers in LANs	Traditional phone lines, early internet connections	Modern internet backbones, long-distance data transmission



Your notes

© 2026 Save My Exams, Ltd.

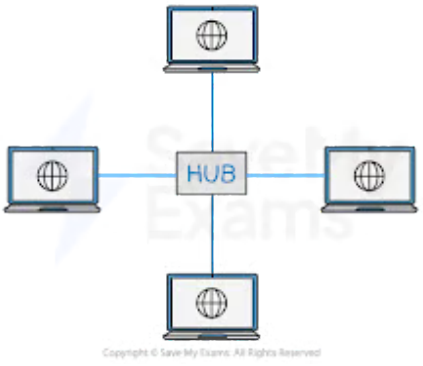
Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

LAN hardware

Hub

- A hub is a networking device which is used to **connect multiple devices in a network**
- Hubs are "**dumb**" devices that pass on anything received on one connection to **all other connections**
- Because all data is sent to all devices, it can lead to **network inefficiencies** and **security issues**
- Hubs allow **multiple** other devices to be connected to them



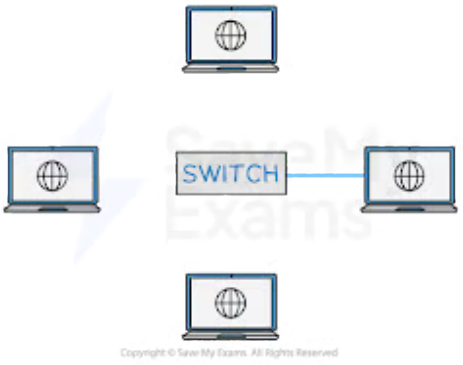
- Hubs are generally **much cheaper** than switches, but:
  - When a hub receives a data packet, it will **broadcast it to every device** on the network
- This creates two potential issues:
  - As the information is being broadcast to every device, it will make **unnecessary traffic**, especially if there are a large number of devices
  - As every device will receive the data packet, **security may be a concern**

Switch

- A network switch is a networking device that **connects devices on a computer network** and uses **packet switching** to receive, process and forward data to the destination device
- Unlike a hub, a switch **only sends data to the device it was intended for**, which improves network efficiency

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com



Your notes

- This is done by each switch having a **lookup table**

Port	Mac address
1	DF-42-B2-11-4D-E3
2	11-14-F2-1D-C3-C6
3	00-4B-17-7C-A2-C9

- When a switch receives a data packet, it **examines the destination MAC address** and **looks up that address in its lookup table**
- Once it has found the matching MAC address, it will then **forward the data packet to the corresponding port**



Server

- A **server** is a powerful computer that provides **services or resources** to other devices on a network (called clients)
- Can **manage and store files**, **host websites**, **control access to printers**, or **run applications**
- Often kept in **dedicated rooms** or data centres due to their importance and high uptime requirements

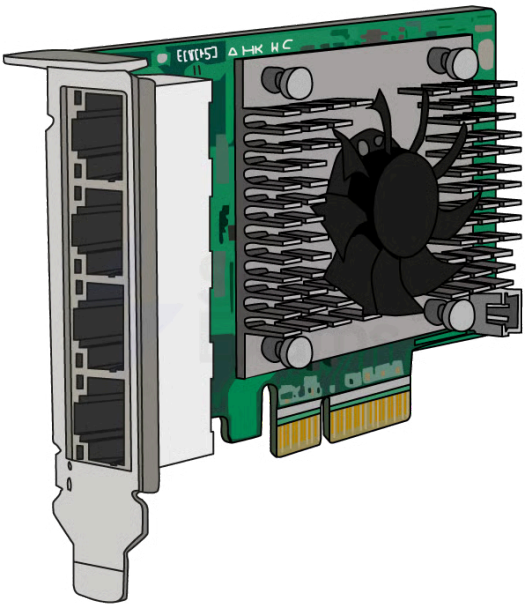
Study copy distributed by EduResource ZA | eduresourcza.com

- Designed to handle **multiple requests at once** and stay on **24/7**
- Servers usually run **specialist operating systems** (e.g. Windows Server, Linux)
- Can be part of a **LAN** (local server) or accessed remotely over a **WAN** (cloud server)
- Examples include: **file server, print server, web server, mail server**

Your notes

Network Interface Card (NIC)

- Historically a card **inserted into a slot on the motherboard** but now more likely to be **built into the motherboard**, that **enables a device to connect to a network**
- NICs have a **built-in ethernet port** and can be connected to a network via an **ethernet cable**
- It provides a **dedicated, full-time connection to a network**, converting the computer's data into a network-friendly format
- Every NIC has a **unique identifier called** a **MAC address**, used to identify the device on the network
- The primary function of a NIC is to **send and receive data packets** between the computer or device and the network


Copyright © Save My Exams. All Rights Reserved

Wireless Network Interface Card (WNIC)

- Allows a device to connect to a **wireless network (Wi-Fi)**
- Often **built into the motherboard** of modern laptops, tablets, and smartphones
- Uses **radio waves** to send and receive data to/from a **wireless router or access point**
- Provides a **dedicated connection** to the network, without needing physical cables

Study copy distributed by EduResource ZA | eduresourcza.com

- Supports standard wireless protocols like **Wi-Fi 4, 5, or 6** depending on the model
- Has a unique **MAC address** used to identify the device on the network
- Converts the device's data into a **wireless signal** suitable for network transmission

Your notes

Wireless Access Point (WAP)

- Allows **wireless devices** to connect to a **wired network**
- Acts like a **bridge** between the wired and wireless parts of a network
- Commonly found in **homes, schools, and offices** as part of a Wi-Fi setup
- Often **built into wireless routers**, but can also be a **separate device** in larger networks
- Uses **radio signals** to send and receive data from wireless devices (e.g. laptops, phones)
- Extends the **range of the wireless network**, especially in large buildings
- Supports communication using standard **Wi-Fi protocols** (e.g. 802.11ac, 802.11ax)



Copyright © Save My Exams. All Rights Reserved

Bridge

- Connects **two separate network segments** together
- Often used to join **two LANs** so they act as one larger network
- Operates at the **data link layer (Layer 2)** of the OSI model
- Can **filter traffic** by checking MAC addresses to decide if data should cross the bridge
- Helps **reduce network traffic** by only forwarding necessary data
- Commonly used in older or more complex network setups

Repeater

- Used to **boost or regenerate signals** in a network
- Helps extend the range of a **wired or wireless signal**
- Receives a weak signal and **retransmits it at full strength**

Study copy distributed by EduResource ZA | eduresourcza.com

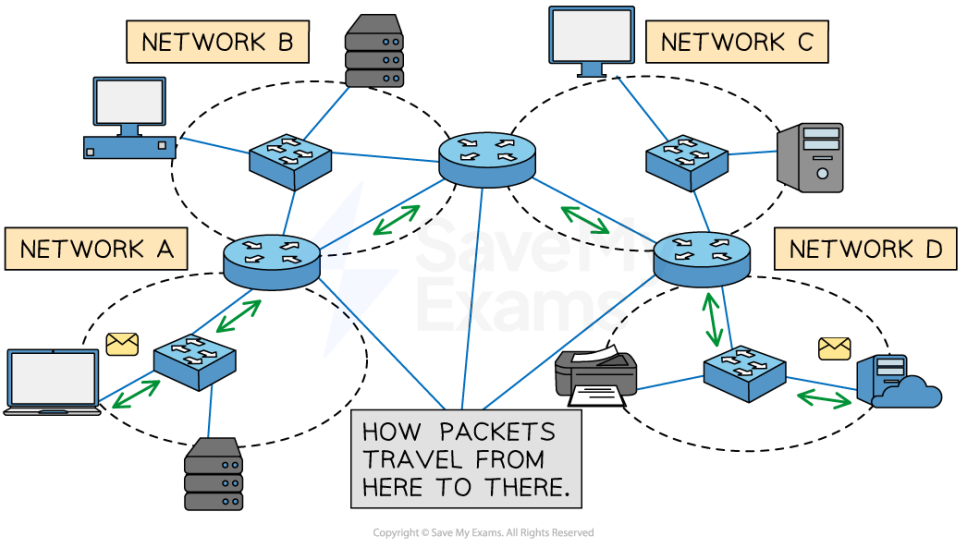
- Used when data needs to travel **long distances** without losing quality
- Often used in **Wi-Fi range extenders** to improve coverage in large buildings
- Operates at the **physical layer (Layer 1)** of the OSI model

Your notes

Router

What is the role of a router?

- A router is a network hardware device that routes data from a local area network (**LAN**) to another network connection - **it joins two networks together**
- Routers **analyse data packets** and **determine the best path** for the packet to reach its destination
  - The **header** contains information about the packet
  - The **payload** is the actual data being sent
  - The **IP address** of both the sender and intended recipient is stored in the header of the data packet
- The router can **often feature additional functionalities** such as wireless networking, built-in firewalls for enhanced security, and network switch capabilities
- A router being used to **connect a LAN to a WAN** will have a public **IP address**, which has been assigned to it by an Internet Service Provider
- It is this public IP address that other routers use to **identify and direct packets** to the network



Multiple networks connected by routers, represented by the blue circular objects

- If the **data packet** is coming into the LAN, the router will send the data packet to the specific device within its LAN that the packet is meant for

Study copy distributed by EduResource ZA | eduresourcza.com


Your notes

- If the packet is being sent from a device within the LAN, it will **read the header** of the packet to **determine the intended destination** IP address
  - It might have to **travel through several routers** before it gets to its destination
  - Each pass from router to router is called a **hop**
  - It will then **forward** the packet to its destination
- The network access device or '**home hub**' used in your home network will have a router **built into it**

Step	Description
1	A router receives incoming data packets from one network and analyses the packet header to determine the destination IP address
2	It then looks up the IP address in a routing table (routing table of known networks) to determine the next network where the packet should be sent
3	The router then forwards the packet to the appropriate network or device

- Every router **repeats this process** the data packet passes through until it reaches its destination
- In addition to routing data between networks, routers can also perform other functions such as:
  - **Assigning IP addresses to devices within the LAN**
  - **Filtering incoming traffic based on certain criteria, such as IP address, port number, or protocol type**

© 2026 Save My Exams, Ltd.

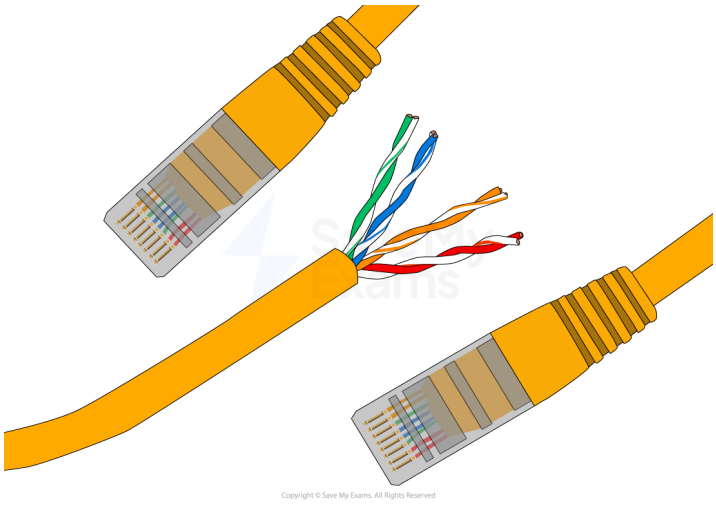
Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Ethernet

What is Ethernet?

- Ethernet is a **protocol** that controls
  - **Wiring**
  - **Data transmission**
  - **Data encapsulation**
- A wired networking **standard** used in a **Local Area Networks (LANs)**
- Data is transmitted in **frames**
- Uses **CSMA/CD** to help detect and avoid data collisions



Ethernet frames

- Data is **transmitted across the network** in structured units called frames
- Each frame typically includes:
  - **Preamble** – A sequence of bits used to **synchronise communication** between devices
  - **Destination MAC Address** – The **unique ID** of the device the data is being sent to
  - **Source MAC Address** – The **unique ID** of the device sending the data
  - **EtherType / Length** – Indicates the **type of data** or the **size** of the payload
  - **Payload** – The **actual data** being sent (e.g. a web page, file, or message)

Study copy distributed by EduResource ZA | eduresourcza.com

- **FCS (Frame Check Sequence)** – Used for **error detection** to check if data arrived correctly



Your notes

CSMA/CD

What is CSMA/CD?

- CSMA/CD is a protocol used to **detect and prevent collisions** in a **bus topology**
- Stands for **Carrier Sense Multiple Access with Collision Detection**
- Devices **check if the channel is free** before sending data
- If the **channel is busy**, the device **waits**
- If the **channel is free**, the device **transmits the data**
- Since all devices share the same cable, **two can send data at the same time**, causing a **collision**
- When a collision happens:
  - Devices **detect it**
  - A **jamming signal** is sent to alert other devices
  - Both devices **stop transmitting**
  - Each device waits for a **random amount of time**, then **tries again**



Worked Example

Ethernet uses Carrier Sense Multiple Access/Collision Detection (CSMA/CD).

Describe CSMA/CD. **[4]**

Answer

1 mark per point to max 4

- A workstation / node (wishing to transmit) listens to the communication channel [1 mark]
  - the data is only sent when the channel is free [1 mark]
- As there is more than one computer connected to the same transmission medium [1 mark]
  - two workstations can start to transmit at the same time, causing a collision [1 mark]
- If a collision happens, the workstations send a jamming signal to abort transmission [1 mark]
  - and each waits a random amount of time before attempting to resend [1 mark]

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Bit streaming

What is bit streaming?

- Bit streaming is the **continuous flow of data (bits)** sent over the internet or a network
- Commonly used for **real-time services** like **video, audio, or live broadcasts**
- Data is sent in **small chunks** so it can be **played back immediately**
- Reduces the need to **download the entire file before use**
- Bit streaming can be categorised into two types:
  - **Real-time**
  - **On-demand**

Real-time vs on-demand

Feature	Real-time streaming	On-demand streaming
What is it?	Watching something **live as it happens**	Watching something **whenever you choose**
Examples	Live sports, live news, online gaming, video calls	Netflix, YouTube, Amazon Prime
Timing	**Streamed and watched at the same time**	**You choose when to watch**
Delays (latency)	Needs **low delay** to feel live	Can handle **small delays** thanks to buffering
Internet needed	Needs a **strong and steady connection**	Can adjust quality based on your connection (adaptive bitrate)
Playback controls	Usually **no pause or rewind** during live events	You can **pause, rewind, or skip** as needed

Factors affecting bit streaming

What factors can affect bit streaming?

- There are **two** main factors to consider when using bit streaming, they are:

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- **Bit rate**
- **Broadband speeds**



Your notes

Bit rate

- Bit rate is the **amount of data able to be transmitted in a specified unit of time** (usually seconds)
- Streaming platforms will **adjust bit rates** based on network performance

Feature	Higher bit rate	Lower bit rate
Quality	Allows for **higher quality** content (HD / 4K)	**Lower quality** – more pixelation or blurriness
Visual & audio clarity	**Sharper and clearer** images and sound	May look **blurry**, especially during fast movement
Bandwidth usage	Uses **more network bandwidth**	Uses **less bandwidth**
Buffering	May require a **larger buffer** to maintain quality	**Smaller buffer** can be enough, but more frequent buffering likely on poor connections

Broadband speeds

- Broadband speeds affects **how well data can be streamed**
- Faster speeds = **better quality** and **smoother playback**

Aspect	High-speed broadband	Slow/restricted broadband
Network capacity	Handles **higher bit rates**	Can't support **high bit rates**
	Allows **multiple devices** to stream smoothly	**Limited bandwidth** affects overall network performance
	Supports **HD/4K quality**	May force streams to use **lower resolution**
User experience	**Smoother streaming** with minimal buffering	**Frequent buffering** and interruptions
	Clear visuals and audio	**Pixelation and lower sound quality**

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

	Ideal for **real-time content** (e.g. gaming, video calls)	Struggles with **live or interactive services**



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Internet vs WWW

What is the Internet?

- The Internet is a **global network of networks** (Interconnected Network)
- The Internet is the most well-known Wide Area Network (**WAN**)
- The Internet is the **infrastructure** used to **provide connectivity** to the World Wide Web (WWW)
- **Uses protocols** like **TCP/IP** to transfer data between devices and networks

What is the World Wide Web (WWW)?

- The world wide web, or simply **the web**, is a **collection of websites** and **web pages** that are **accessed using the Internet**
- It was created in 1989 by **Tim Berners-Lee**, who envisioned it as a way to **share** and **access** information **on a global scale**
- The web consists of **interconnected documents** and **multimedia files** that are stored on **web servers** around the world
- **Web pages** are accessed using a web **browser**, which communicates with web **servers** to retrieve and display the content
- **Uses protocols** like **HTTP and HTTPS** to request and deliver web content

Internet hardware

Modem

- **Modem** stands for **modulator-demodulator**
- Converts **digital signals** from a computer into **analogue signals** to send over telephone or cable lines
- Also **converts analogue signals back into digital** so the computer can understand them
- Allows devices to **connect to the internet** through technologies like:
  - **DSL**
  - **Cable**
  - **Dial-up**
- Used to **send and receive data** over long distances via traditional communication lines

Public Switched Telephone Network (PSTN)

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- Developed soon after the **invention of the telephone** in the late 1800s
- Originally built to handle **voice communication** over copper wires
- As the demand for other forms of communication grew, the PSTN was **adapted to enable internet access**
- Used to **connect computers/devices and LANs** between **towns** and **cities**
- Traditional **copper telephone lines** are being slowly replaced with **fibre optic cables**
  - Fibre gives access to **higher bandwidth** and **faster data transfer speeds**


Your notes

Dedicated lines

- Dedicated lines (e.g. **T1**, **T3**, and **fibre-optic**) provide **exclusive, high-speed connections** between two points
- Unlike the **PSTN**, these connections are **always active** and **not shared** with other users
  - **Faster** and **more reliable**
  - Ideal for **businesses** and **organisations** needing consistent, high-bandwidth internet
  - Better for activities like **video conferencing**, **cloud computing**, and **large file transfers**

Connection type	Speed
T1 Line	Up to **1.54 Mbps**
T3 Line	Up to **45 Mbps**
Fibre-Optic	Speeds in the **gigabits**

Cellular networks

- Provide **wireless communication** for mobile phones and other portable devices
- The network is made up of **cells**, each served by a **cell tower (base station)**
- Devices connect to the **nearest tower** to send and receive signals
- As a user moves, their device **automatically switches** to the next closest tower (handover)
- Used for **voice calls**, **text messaging**, and **mobile data (internet access)**
- Support different generations of mobile technology:
  - **2G** – Basic calls and texts
  - **3G** – Mobile internet and video calling

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- **4G** – Fast internet browsing and streaming
- **5G** – Very high-speed internet, low latency (great for gaming and real-time apps)



Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

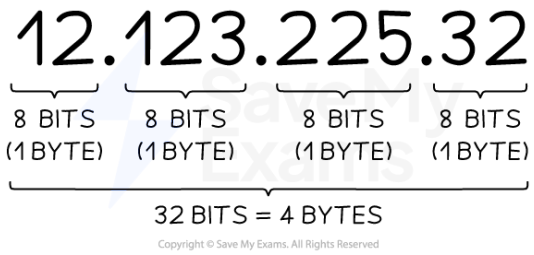
IP addressing

What is an IP address?

- An IP (Internet Protocol) address is a **unique identifier** given to devices which communicate over the Internet (**WAN**)
- IP addresses are **dynamic**, they can change
- IP addresses make it possible to **deliver data to the right device**
- A device connecting to a network **will be given an IP address**, if it moves to a different network then **the IP address will change**

IPv4

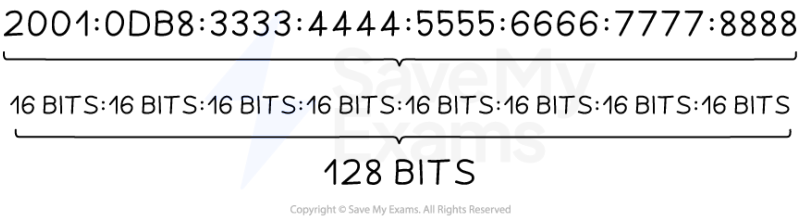
- Internet Protocol version 4 is represented as **4 blocks of denary numbers** between **0** and **255**, separated by **full stops**
- Each block is **one byte** (8 bits), each address is **4 bytes** (32 bits)



- IPv4 provides over **4 billion unique addresses** (2^{32}), however, with over **7 billion** people and **countless devices per person**, a solution was needed

IPv6

- Internet Protocol version 6 is represented as **8 blocks of 4 hexadecimal digits**, separated by **colons**
- Each block is **2 bytes** (16 bits), each address is **16 bytes** (128 bits)



Study copy distributed by EduResource ZA | eduresourcza.com

- IPv6 could provide over **one billion unique addresses** for **every person** on the planet (2^{128})



Your notes

Subnetting

- Subnetting is the process of **dividing a larger network into smaller, more manageable parts**, called **subnets** (short for sub-networks)
- Each subnet works like a **mini-network** within the main network, allowing devices to communicate more efficiently
- Benefits of subnetting include:
  - **Reduces network traffic** – less data is broadcast across the whole network
  - **Improves speed and performance** – data stays within its **local subnet**
  - **Increases security** – limits access so **not all devices can reach all parts** of the network
  - **Easier to manage and maintain** – changes can be made to **one subnet** without affecting the rest
  - **Improves organisation** – helps group devices by **department or function**
- It's commonly used in **larger networks**, like schools or businesses, to reduce traffic, keep data local, and make management easier

Public vs private IP addresses

- Public IP addresses are assigned to devices that need a **constant connection to the internet**
- Examples:
  - **Web servers**
  - **Email servers**
- **Globally unique** – no two devices can have the same public IP
- Allows devices to be **directly accessed** from anywhere on the internet
- Private IP addresses are assigned to devices on a **Local Area Network (LAN)** by a **router**
- **Not routable on the internet** – improves **network security**
- Used for items in a home or office, such as:
  - **Laptops**
  - **Phones**
  - **Printers**
- Allows **internal communication** without **exposing devices to the public** internet

Static vs dynamic IP addresses

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- Static IP addresses are **fixed IP addresses** that **do not change**
- Assigned to devices that need a **consistent address**

  Commonly used for:

  - **Websites**
  - **Remote access services**
  - **Email or file servers**
- **No management required** once set
- Allows **reliable access** from anywhere on the network or internet
- Dynamic IP addresses are **temporarily assigned** when a device connects to the network
- Comes from a **pool of available IP addresses**
- Managed automatically by a **DHCP server (Dynamic Host Configuration Protocol)**
- Ideal for devices where a **fixed address isn't needed**
  - e.g. laptops, smartphones, or guest devices



Worked Example

Computers on the Internet have IP addresses.

Describe the format of an IP address. **[3]**

Answer

IPv4

- Four groups of denary or hexadecimal digits [1 mark]
- Numbers between 0 and 255 / 0 and FF [1 mark]
- Each is stored in 1 byte / 8 bits [1 mark]
- The whole number is stored in 32 bits / 4 bytes [1 mark]
- Separated by full stops [1 mark]
- Correct example [1 mark]

OR

IPv6

- Eight groups of (Hexadecimal) digits [1 mark]
- Numbers between 0 and FFFF [1 mark]
- Each is stored in 2 bytes/16 bits [1 mark]
- The whole number is stored in 128 bits / 16 bytes [1 mark]
- Separated by colons [1 mark]
- The first instance of multiple groups of zero can be replaced by a double colon [1 mark]
- Correct example [1 mark]



Your notes

Study copy distributed by EduResource ZA | eduresourcza.com



Your notes

URL

What is a URL?

- A **Uniform Resource Locator** (URL) is a **unique identifier** for a web page, known as the **website address**
- It is **text-based** to make it easier to remember
- A user **enters a URL** into a web browser to view a web page
- An example of a URL is:
  - https://www.savemyexams.com/a-level/computer-science/
- A URL can typically be split into three parts:
  - **Protocol**
  - **Domain name**
  - **Web page/file name**
- Using the example about the URL would be split as follows:

Protocol	https	Communication method to transfer data between client and server
Domain name	www.savemyexams.com	Name of the server where the resource is located
Web page/file name	/a-level/computer-science/	Location of the file or resources on the server

DNS

What is the Domain Name System (DNS)?

- The **DNS** is like the **Internet's phone book**
- It **translates domain names** (like www.google.com) into **IP addresses** (like 142.250.180.68)
- Computers use IP addresses to **find and connect to servers**
- When you type a URL into your **browser**:
  - DNS **finds the matching** IP address
  - Your device **connects to that server** to load the website

© 2026 Save My Exams, Ltd.

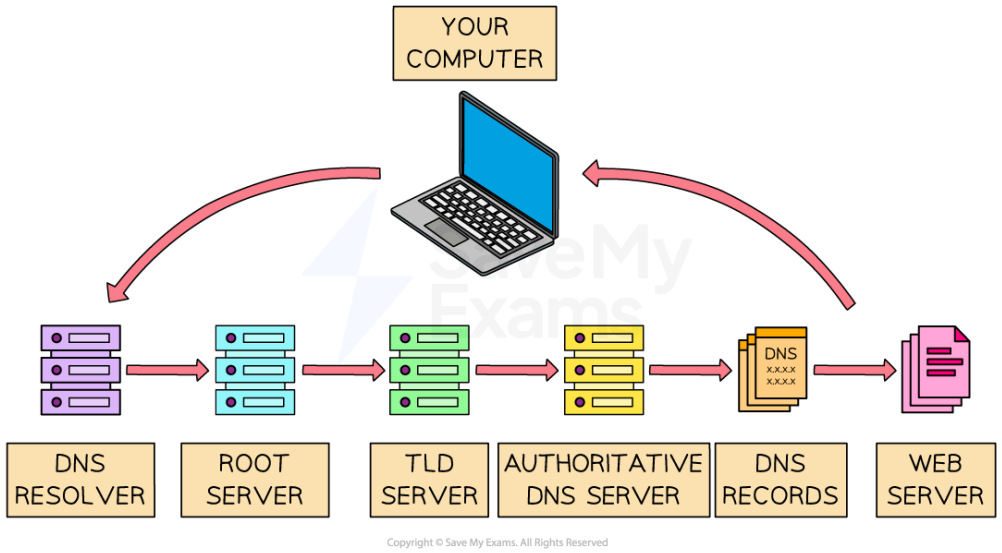
Study copy distributed by EduResource ZA | eduresourcza.com

- Without DNS, users would have to **remember the IP address** of every website they visit
- When a **domain is registered** or its server **IP address changes**, the DNS must be updated
  - This update is called **DNS propagation**, and it may take some time to spread across the internet

Your notes

What happens when you type a URL into a web browser?

1. The user enters a **URL** into the address bar of the **web browser**
2. The browser checks its **cache** to see if it already knows the IP address for the website
3. If not found, it sends the domain name to a **DNS** server, which stores an **index** of domain names and their matching IP addresses
4. If the DNS server finds the IP address, it sends it back to the **web browser**
5. If it does not find the IP address, it passes the request on to a **higher-level DNS server**
6. This may involve contacting:
   - A **root server**, which points to the correct **Top-Level Domain (TLD) server** (e.g. .com, .org),
   - The **TLD server**, which points to the **authoritative DNS server** for the domain
7. The **authoritative DNS server** responds with the correct IP address
8. The web browser then sends a **request** to the web server at that IP address
9. The **web server** processes the request and sends back the website's data (such as HTML, images, CSS, and JavaScript)
10. Finally, the **web browser renders** the content and displays the website to the user



Study copy distributed by EduResource ZA | eduresourcza.com

Worked Example

Complete the following paragraph that describes the sequence of events when a user requests a page from a website.

1. The user enters the Uniform Resource Locator (URL) into the address bar of the ..... .
2. The domain name is taken from the URL and sent to a Domain Name Service (DNS). This stores ..... of domain names and their matching IP addresses.
3. If it finds the domain name, it sends the IP address to the ..... .
4. If it does not find the domain name, it sends the request to .....

Answer

1. Web browser [1 mark]
2. An index/a list/a table [1 mark]
3. Web browser [1 mark]
4. A higher-level DNS [1 mark]



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Computers & Components

Contents

- System components
- Embedded systems
- Hardware devices
- Memory types
- Monitoring & control

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Input/output devices

What is an input device?

- Input devices are **hardware components that allow users to interact** with a computer system
- They **enable the user to input data or commands** into the system, which the computer then processes to produce output

Types of input devices

Name of device	Description
Keyboard	The most common input device. It allows users to input text and commands by pressing keys
Mouse	Allows users to navigate the computer screen, click on items, and perform other functions
Touchscreen	Found on smartphones, tablets, and some computers, touchscreens allow users to interact with the device by touching the screen
Microphone	Captures audio input, which can be used for voice commands, recording audio, or video conferencing
Webcam	Captures video input, often used for video conferencing or recording videos
Scanner	Digitises physical documents or images, converting them into a format that the computer can process
Game Controller	Used primarily for video games, these devices allow users to control game characters and interact with the gaming environment
Graphics Tablet	Allows artists and designers to draw or sketch directly onto a computer. It's particularly useful for graphic design, 3D modelling, and other visual creative tasks
Biometric Devices	These devices, such as fingerprint scanners or facial recognition systems, are used for security purposes to verify a user's identity

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Barcode Reader	Scans barcodes, typically used in retail and inventory management
Joystick	Often used for computer games, especially flight simulators. It allows the user to control movement more fluidly than with a keyboard or mouse



Your notes

What is an output device?

- Output devices are **hardware components that receive information** from a computer system and present it to the user in a comprehensible form
- They enable the computer to **communicate the results of processed data or commands**

Types of output devices

Name of device	Description
Monitor	This is the most common output device. It displays visual output from the computer, including text, images, and videos
Printer	Produces a hard copy of digital documents or images. There are various types of printers, including inkjet, laser, and 3D printers
Speakers	Output audio from the computer, such as music, sound effects, or voice
Headphones	Similar to speakers, headphones output audio directly to the user, providing a more personal and potentially immersive experience
Projector	Projects the computer's display onto a large screen or wall, useful for presentations or movie viewing
Braille Display	This specialised device outputs information in Braille, allowing visually impaired users to read text from the computer
Plotter	Used for printing large, high-quality diagrams and designs, often used in engineering or architecture
Virtual Reality (VR) Headset	Provides an immersive visual and audio output, primarily used for gaming and virtual simulations
Computer-Controlled Machinery	In manufacturing or robotics, computers often output commands directly to machinery to control their operation

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Choosing the right device



Your notes

When recommending a device for a specific situation, consider the following factors:

- **User Needs**
  - What tasks will the user be performing? A graphic designer might need a graphics tablet, while a data entry clerk might need a keyboard with a number pad
- **User Skills**
  - Is the user comfortable with the device? A touch screen might be more intuitive for some users, while others might prefer a mouse and keyboard
- **Environment**
  - Where will the device be used? A wireless mouse might be suitable for a clutter-free office, while a wired mouse might be better for a public computer lab to prevent theft
- **Cost**
  - Higher-end devices often have more features but are also more expensive. Consider the budget and whether the extra features are worth the cost

Primary memory

What is primary memory?

- Primary memory is **memory directly accessible by the CPU**
- Has **much faster access times** than secondary storage
- This **speeds up operations** like the **Fetch-Execute Cycle**
- Stores **data and instructions** the CPU needs while the computer is **on and running**
- Acts as **short-term, working memory**
- Found in components like:
  - **RAM** (Random Access Memory) – directly connected to the CPU
  - **Cache and Registers** – built **into the CPU** for even faster access
- Because it's fast, it's also **more expensive**, so we use **less of it**
  - Example: RAM = **16–32 GB**
  - Secondary storage (like HDDs) = **1–2 TB or more**
- View more detailed information on the different types of memory

Secondary storage

What is secondary storage?

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com



Your notes

- Secondary storage provides **permanent data storage**
- Hardware components that **retain digital data** within a computer system
- They provide a means of **storing**, **accessing**, and **retrieving** data, which can include software applications, documents, images, videos, and more
- There are 3 types of storage:
  - **Magnetic**
  - **Optical**
  - **Solid state**

Type of storage	Description	Benefits	Drawbacks
Magnetic (e.g., Hard Disk Drives, Magnetic Tape Drives)	Store data by magnetising particles on a disk or tape	High storage capacity; relatively low cost per gigabyte; suitable for long-term storage and backup	Slower read/write speeds compared to other types; susceptible to physical damage; moving parts can wear out over time
Optical (e.g., CDs, DVDs, Blu-ray Discs)	Store data using a laser to burn pits into the surface of the disc	Durable and relatively immune to environmental conditions; easy to transport; suitable for distributing software, music, or movies	Lower storage capacity compared to other types; slower read/write speeds; can be easily scratched or damaged
Solid state (e.g., Solid-State Drives, USB Flash Drives)	Store data in flash memory cells	Fast read/write speeds; no moving parts, so less likely to fail due to physical shock; silent operation	Higher cost per gigabyte; flash memory cells can wear out after a certain number of write cycles

Here are some of the **devices commonly used** for storage:

Name of device	Type of device	Typical storage capacity	Affordability	Portability	Durability
Hard Disk Drive (HDD)	Magnetic	500GB - 2TB (consumer-grade)	Low cost per GB	Low (especially for internal HDDs)	Moderate (susceptible to damage from shocks or falls due to moving parts)

Study copy distributed by EduResource ZA | eduresourcza.com

Solid-State Drive (SSD)	Flash	120GB - 4TB (consumer-grade)	High cost per GB	High (especially for external SSDs)	High (no moving parts, less susceptible to physical shock)
USB Flash Drive	Flash	8GB - 256GB (common sizes)	Moderate cost per GB	Very High (small and lightweight)	Moderate (can withstand casual handling, but can be lost or damaged if not cared for)
CD/DVD/Blu-ray Disc	Optical	CD: 700MB, DVD: 4.7GB - 9GB, Blu-ray: 25GB - 50GB	Low cost per disc	High (thin and lightweight)	Low (can be scratched or damaged easily)



Your notes

Choosing the right storage device

When recommending a storage device for a specific situation, consider the following factors:

- **Storage needs**
  - How much data does the user need to store? A user with large amounts of data might need a high-capacity HDD, while a user who only needs to store a few documents might be fine with a USB flash drive
- **Performance needs**
  - Does the user need fast access to their data? An SSD might be best for tasks that require high-speed data access, like video editing or gaming
- **Portability**
  - Does the user need to transport the data? USB flash drives and external HDDs or SSDs are portable and can be used to transfer data between different computers
- **Cost**
  - Higher-capacity and faster storage devices are generally more expensive
  - Consider the user's budget and whether their storage and performance needs justify the extra cost

© 2026 Save My Exams, Ltd.

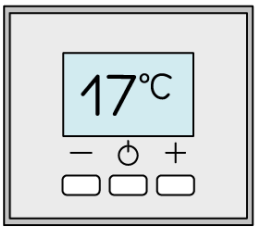
Study copy distributed by EduResource ZA | eduresourcza.com

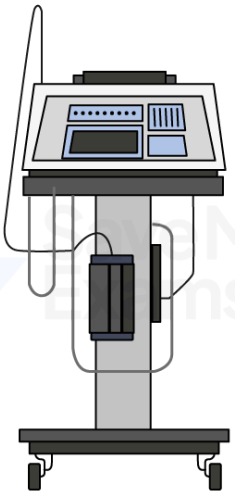
Your notes

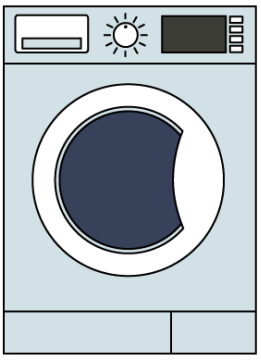
Embedded systems

What is an embedded system?

- An embedded system is a computer system which is used to **perform a dedicated function**, inside a larger mechanical unit
- Examples of embedded systems include
  - **Heating thermostats**
  - **Hospital equipment**
  - **Washing machines**
  - **Dishwashers**
  - **Coffee machines**
  - **Satellite navigation systems**
  - **Factory equipment**
  - **Security systems**
  - **Traffic lights**







Copyright © Save My Exams. All Rights Reserved

Benefits and drawbacks of embedded systems

Benefits	Drawbacks

Study copy distributed by EduResource ZA | eduresourcza.com

Small and compact – easy to fit into dedicated devices	**Limited functionality** – only performs specific tasks
Low power consumption – efficient and cost-effective	**Hard to upgrade or repair** – often built into the device
Fast and reliable – designed for quick, repetitive tasks	**Limited memory and processing power**
Cheaper to produce – uses minimal hardware	**Not flexible** – can't easily be reprogrammed for other tasks
Works in real-time – ideal for time-sensitive operations (e.g. alarms)	**May be less secure** – limited protection if connected to other systems



Your notes



Worked Example

A car has several features. One feature is a lane detection system. This system monitors the lines on either side of the lane. If the car gets too close to one line, the system automatically moves the car away from the line.

Explain why the lane detection system is an example of an embedded system. **[2]**

Answer

- The lane detection system is built into / integrated into the car [1 mark]
- The lane detection system only performs one task [1 mark]
- The lane detection system is not easily changed/updated by the car owner [1 mark]

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Input devices

Device	Principal operations	Additional information
Microphone	▪ Converts sound waves into electrical signals	▪ Allows users to record voice or send audio into a computer ▪ **Dynamic microphones** – good for loud environments (e.g. concerts) ▪ **Condenser microphones** – more sensitive and accurate, used in studios
Touchscreen	▪ Detects a user's touch and converts it into an input command	▪ **Capacitive** – responds to electrical charge from your finger (used in smartphones, tablets) ▪ **Resistive** – responds to pressure (used in ATMs, tills) ▪ First used in ATMs and information kiosks, now used in smartphones, tablets, laptops, smart displays ▪ Popular for direct, easy interaction and improved accessibility

Output devices

Device	Principal operations	Additional information
Laser printer	▪ Laser beam draws the image of the page onto a photosensitive drum, changing its electric charge ▪ Toner is transferred from drum to paper	▪ **Toner sticks to the drum** – toner powder is attracted to charged areas matching text/image shape ▪ **Fusing** – paper passes through hot rollers, melting toner onto paper so it doesn't smudge
3D printer	▪ Builds objects layer by layer from the bottom up ▪ Uses various materials such as thermoplastics, resins, and metals	▪ Handles complex shapes traditional methods can't make easily ▪ **FDM** – melts plastic and builds in layers, **SLA** – uses light to harden liquid resin

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

	▪ Allows high customisation and rapid prototyping	▪ Used in healthcare (prosthetics), automotive/aerospace (custom parts) ▪ Can be slow for large/detailed objects, some methods need costly specialist materials
Speakers	▪ Convert electrical signals into sound waves	▪ Range from basic laptop speakers to high-end multi-driver home theatre systems ▪ Found in phones, laptops, studios, smart devices, home theatres ▪ Improved with digital sound processing and miniaturised components ▪ Support voice commands, calls, and multimedia playback
Virtual reality headset	▪ Creates a fully immersive 360° digital environment ▪ Lets users look around and interact with the virtual world	▪ Uses head tracking, motion sensors, and stereoscopic displays for 3D vision ▪ **Uses** – gaming, education/training, architecture/design, medical/therapy ▪ **Challenges** – expensive hardware, possible eye strain or motion sickness, time-intensive content creation



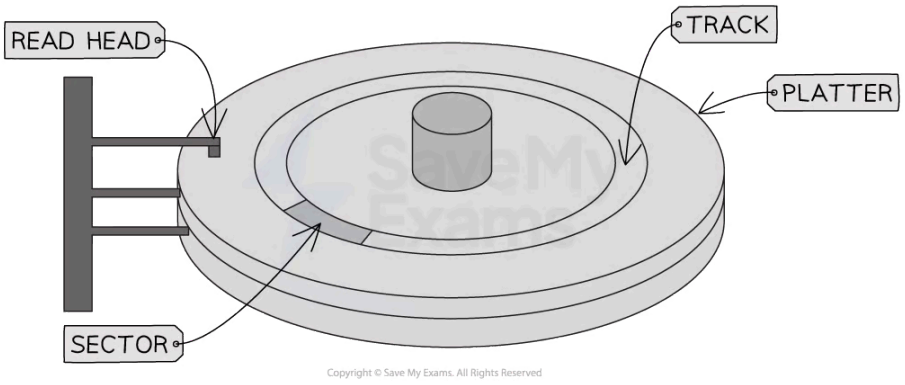
Your notes

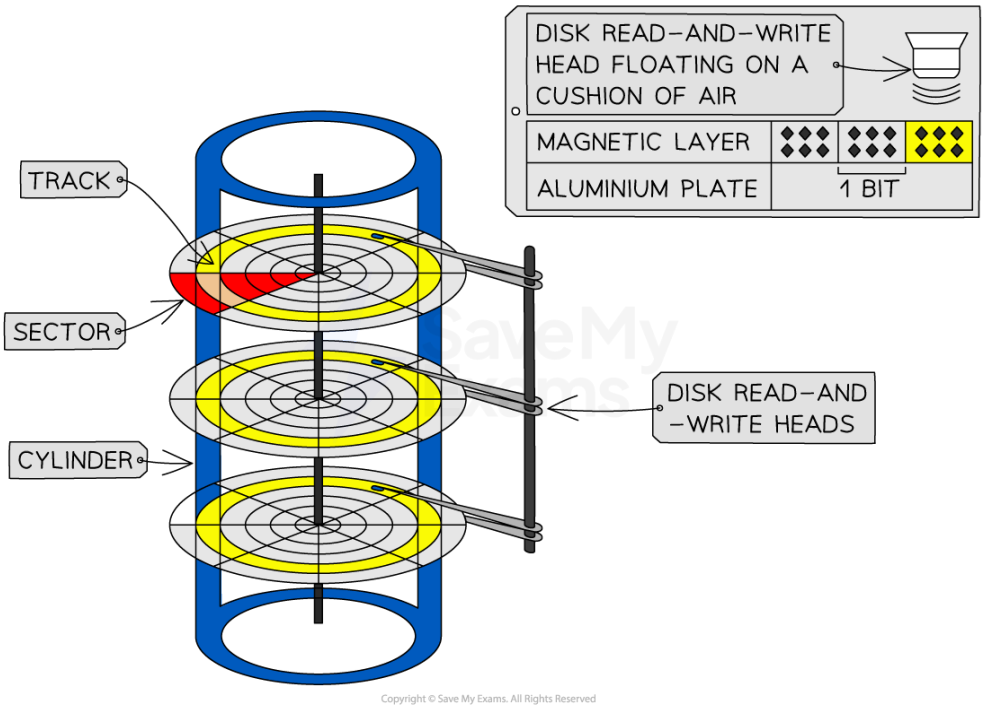
Storage devices

Device	Principal operations	Additional information

Study copy distributed by EduResource ZA | eduresourcza.com

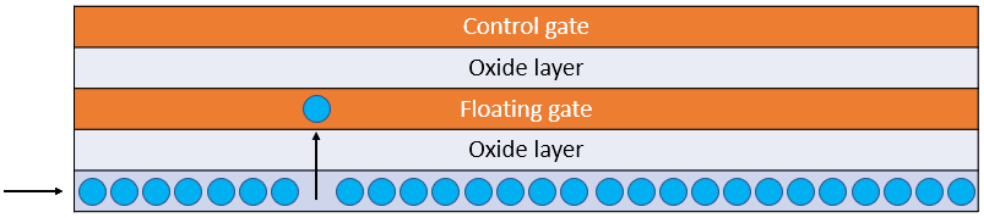
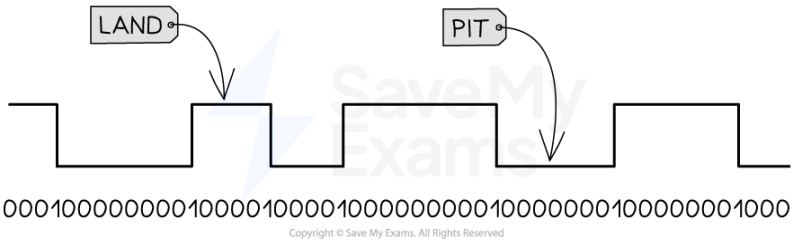
Magnetic hard disk	▪ Multiple metal platters coated with magnetic material store data as magnetised iron particles (0s and 1s) ▪ Platters spin at high speed (typically 5400–7200 RPM) ▪ Data read/written using electromagnets	▪ **Platters** divided into concentric tracks and wedge-shaped sectors, forming track sectors ▪ **Read/write arm**, controlled by an actuator, positions the head over the correct track sector ▪ Reliable but mechanical – can be slower and more prone to damage than SSDs





Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

Solid state (flash) memory	▪ Stores data in **cells** using transistors that act as switches ▪ Uses **NAND** or **NOR** gates to control flow of electrons ▪ Writing: high voltage **pushes electrons** onto floating gate ▪ Faster and more durable than magnetic drives	▪ Common examples: SSDs, USB flash drives ▪ Each cell contains a **control gate** (controls current) and a **floating gate** (stores charge) ▪ Erasing: reverse high voltage pulls electrons off floating gate ▪ No moving parts – better shock resistance
		
Optical disc reader/writer	▪ Uses a **laser** to read and/or write data on optical discs ▪ Writing: **laser burns pits and lands** onto the disc surface ▪ **Arm** moves the laser across the disc ▪ Blu-rays store the most, CDs the least	▪ Examples: CD, DVD, Blu-ray ▪ Reading: laser scans surface; changes in reflection indicate 0s and 1s ▪ CD-R = write once, CD-RW/DVD-RW = rewriteable ▪ Slower than magnetic/solid-state storage but good for archival
		

Your notes

Worked Example

Study copy distributed by EduResource ZA | eduresourcza.com

A magnetic hard disk is used to store data on the computer. Describe the principal operations of a magnetic hard disk **[5]**

Answer

- The hard disk has (one or more) platter/plate/disk [1 mark]
- Each surface of the platter/disk is capable of being magnetised [1 mark]
- The surface of the platter/disk is divided into concentric tracks / circles [1 mark]
- The disks are rotated (at high-speed) [1 mark]
- (Each surface of the disk) has a read/write head mounted on an arm (positioned just above the surface) [1 mark]



Your notes

Study copy distributed by EduResource ZA | eduresourcza.com



Your notes

RAM vs ROM

What is RAM?

- RAM (Random Access Memory) is **primary storage** that is directly connected to the CPU and holds the **data** and **instructions** that are **currently in use**
- RAM is **volatile** which means the contents of RAM are lost when the power is turned off
- For the CPU to access the data and instructions they must be **copied from secondary storage**
- RAM is **very fast** working memory, much faster than secondary storage
- RAM is **read/write** which means data can be read from and written to
- In comparison to ROM, it has a much **larger capacity**

What is ROM?

- ROM (Read Only Memory) is **primary storage** that holds the **first instructions** a computer needs to **start up** (Bootstrap)
- ROM contains the **BIOS** (Basic Input Output System)
- ROM is a small **memory chip** located on the computers **motherboard**
- ROM is **fast** memory, much faster than secondary storage but **slower than RAM**
- ROM is **non-volatile** which means the contents of ROM are not lost when the power is turned off
- ROM is **read only** which means data can only be read from
- In comparison to RAM, it has a much **smaller capacity**

Differences between RAM & ROM

Feature	RAM	ROM
Speed	Very fast	Fast (slower than RAM)
Capacity	Gigabytes (GB)	Megabytes (MB)
Stores	Programs and data in use	Bootstrap (start-up instructions)
Read/Write	Read & write	Read only

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Volatile/Non-volatile	Volatile	Non-volatile



Your notes

SRAM vs DRAM

What is SRAM?

- SRAM (Static RAM) is a form of RAM that **keeps data** as long as power is on
- SRAM is made from **flip-flops** so there is no need for constant refreshing
- SRAM is used in places where **speed** is more important than storage size
- An example of where SRAM is used is:
  - **Cache memory**, where quick access to data is most important
- **Very fast** – faster than DRAM
- **Uses less power**, good for **low-power devices**
- **Expensive** to make
- **Takes up more space** – lower storage capacity compared to DRAM

What is DRAM?

- DRAM (Dynamic RAM) is a form of RAM that stores each bit in a **tiny capacitor**
- DRAM needs **constant refreshing** to keep the data
- DRAM is commonly used as:
  - **Main memory (RAM)**, where larger amounts of cheaper storage is required
- **Cheaper** to produce than SRAM
- **Higher capacity** – can fit **more memory in less space**
- **Slower** than SRAM, needs time to refresh data
- **Uses more power**, especially during refreshing cycles

PROM vs EPROM vs EEPROM

What is PROM, EPROM & EEPROM?

- PROM, EPROM and EEPROM are all **types of ROM** that are **programmed and reprogrammed in different ways**
- Each type has a **specific application** in difference devices

Feature	PROM	EPROM	EEPROM

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Can be reprogrammed?	No – programmed once only	Yes – can be erased and rewritten	Yes – can be erased and rewritten
Erased using	Cannot be erased	UV light	Electric voltage
Must be removed from device?	No	Yes – must be removed from the device	No – can be erased in place
Erased all at once?	Not applicable	Yes – entire chip is erased	No – specific parts can be erased
Common use	Permanent firmware	Reprogrammable chip development	Flash memory, BIOS chips
Examples	Remote controls, basic calculators, early model washing machines	Arcade machines (older models) Early games consoles	BIOS chips in computers, Smart cards, remote key fobs, flash memory like USB sticks and SSDs

Your notes

Worked Example

A computer has hardware and software.

The hardware includes different types of memory.

Complete the description of computer memory. **[5]**

Random Access Memory (RAM) and Read Only Memory (ROM) are both examples of ..... memory.

One item that is stored in RAM is ..... . One item that is stored in ROM is ..... . RAM can be either Static RAM (SRAM) or Dynamic RAM (DRAM).

SRAM uses transistors arranged as ..... . DRAM uses transistors and .....

Answer

Random Access Memory (RAM) and Read Only Memory (ROM) are both examples of primary memory.

One item that is stored in RAM is currently running software/data/part of the OS.

One item that is stored in ROM is the start-up/boot-up instructions/BIOS.

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

RAM can be either Static RAM (SRAM) or Dynamic RAM (DRAM).

SRAM uses transistors arranged as flip-flops.

DRAM uses transistors and capacitors.



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Monitoring vs control

Monitoring systems

- A monitoring system is used to **collect data continuously** through observation
- It works by **passively gathering data**
- It **does not interact with or change** the environment
- The system **does not take action** based on the data collected
- Designed for **high accuracy** using **precise sensors and measurements**
- Examples of monitoring systems include:
  - **Weather stations**
  - **Patient monitoring in hospitals**

Weather stations

- Collect data like **temperature, wind speed, humidity, and rainfall**
- Used by **meteorologists** to observe and predict weather patterns
- Data is collected **24/7** but the system does **not react or change anything**

Patient monitoring

- Tracks **heart rate, oxygen levels, and blood pressure** in real-time
- Alerts medical staff if readings go outside safe ranges
- The system itself just **records and displays data**, it doesn't directly treat the patient

Control systems

- A control system is used to **automatically manage or adjust** a process based on data collected from sensors
- It works by **monitoring input**, then **taking action** if certain conditions are met
- Unlike monitoring systems, control systems **do interact with the environment**
- They are designed to keep systems **stable, safe, or working efficiently** without human input
- Examples of control systems include:
  - **Central heating system**
  - **Automatic irrigation system**

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Central heating system

- **Monitors room temperature** using a thermostat
- If the temperature drops below the set level, the **boiler is switched on**
- Once the target temperature is reached, the system **turns the heating off** automatically
- The system constantly checks and **adjusts** to maintain the desired temperature

Automatic irrigation system

- **Monitors soil moisture levels** in farmland or gardens
- If moisture drops too low, it **activates sprinklers** to water the plants
- Once the soil reaches the correct level, the system **turns off the water supply**
- Ensures plants are **watered efficiently** without wasting resources



Examiner Tips and Tricks

- **Actuators** are output devices in a control system that carry out actions based on data from sensors
- Examples include motors, heaters, valves, and pumps

Use of sensors

What are sensors?

- Sensors are **input devices**
- They **measure a physical property of their environment** such as light levels, temperature or movement
- Sensors can be used for both **monitoring and control systems**
- A process where **outputs are recycled and used as inputs**, creating a continuous cycle is called a **feedback loop**

Sensor type	What it measures	Typical use
Acoustic	Sound levels	To detect changes in sound levels of industrial machinery To monitor noise pollution In security system to detect suspicious sounds



Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

Accelerometer	Acceleration rate, tilt, vibration	Detecting sudden changes in vehicle movement and deploy safety features if needed In mobile phones to detect orientation of the device
Flow	Rate of gas, liquid or powder flow	Detect changes in the flow through pipes in water system
Gas	Presence of a gas e.g. carbon monoxide	Detect levels of gas in confined spaces Detect gas levels when fixing gas leaks
Humidity	Levels of water vapour	Monitor humidity in greenhouses
Infra-red	Detecting motion or a heat source	Security systems detecting intruders who break the beam Measures heat radiation of objects - used by emergency services to detect people
Level	Liquid levels	Detects levels of petrol in a car tank Detect levels of water in a water tank Detect a drop in water levels due to a leak
Light	Light levels	Automatically switching on lights when it gets dark (street lights, headlights)
Magnetic field	Presence and strength	Anti-lock braking system Monitoring rotating machinery such as turbines
Moisture	Presence and levels of moisture	Monitoring moisture in the soil Monitoring dampness in buildings
pH	Acidity or alkaline	Monitoring soil to ensure optimum growing conditions Monitor ph levels in chemical processes
Pressure	Gas, liquid or physical pressure	Monitoring tyre pressure Monitoring pressure in pipes during the manufacturing process



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Proximity	Distance	Monitoring the position of objects in robotics Used in safety systems to prevent objects from colliding
Temperature	Temperature in Celsius, Fahrenheit or Kelvin	Used to maintain temperature in swimming pools Used to control temperature in chemical processes



Your notes

Feedback loops

What is a feedback loop?

- A feedback loop is when a control system uses its **output to influence its next input**
- Allows the system to **automatically adjust** and stay within set conditions
- Feedback allows the system to **check if it's working as expected**
- The **output affects the next input**, helping the system make adjustments
- This means the system can **automatically respond to changes** in its environment
- Helps the system **stay within set limits or target values** (e.g. temperature, moisture)
- Makes the system **more accurate and efficient** without needing human control

Example: Central heating system

- The system uses a **thermostat** to monitor the **room temperature**
- If the room gets **too cold**, the system **turns the heating on**
- Once the room reaches the **set temperature**, the system **turns the heating off**
- This process uses **feedback**:
  - The **output** (room temperature) affects the **input** (whether heating is needed)
- Feedback ensures the system **automatically adjusts** to keep the room at the right temperature
- No need for manual control as the system **self-corrects** using feedback

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Logic Gates & Logic Circuits

Contents

- Logic gates
- Logic construction

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

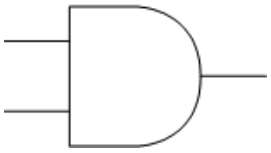
Your notes

Logic gates

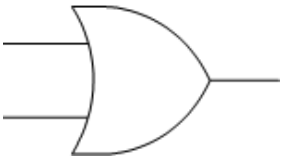
What is a logic gate?

- A logic gate is a **visual way** of representing a **Boolean expression**
- The logic gates covered in this course are:
  - **AND**
  - **OR**
  - **NOT**
  - **XOR**
  - **NAND**
  - **NOR**

AND

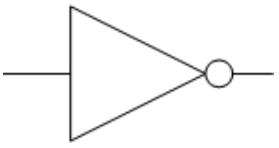
Expression operator	Circuit symbol	Function
(A **AND** B)		Returns TRUE only if **both** inputs are TRUE **TRUE** AND **TRUE** = **TRUE** Otherwise = FALSE

OR

Expression operator	Circuit symbol	Function
(A **OR** B)		Returns TRUE if **either** input is TRUE **TRUE** OR **FALSE** = **TRUE** **FALSE** OR **FALSE** = **FALSE**

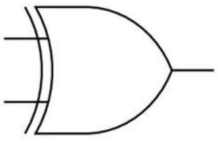
NOT

Study copy distributed by EduResource ZA | eduresourcza.com

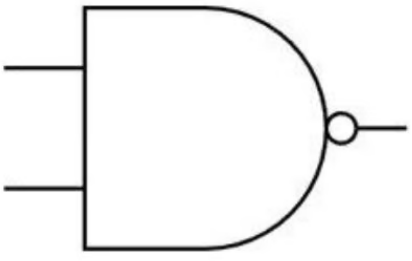
Expression operator	Circuit symbol	Function
(**NOT** A)		**Reverses** the input value NOT **TRUE** = **FALSE** NOT **FALSE** = **TRUE**

Your notes

XOR (exclusive OR)

Expression operator	Circuit symbol	Function
(A **XOR** B)		Returns TRUE if **either** input is TRUE but **NOT** both **TRUE** OR **FALSE** = **TRUE** **FALSE** OR **FALSE** = **FALSE** **TRUE** OR **TRUE** = **FALSE**

NAND (not and)

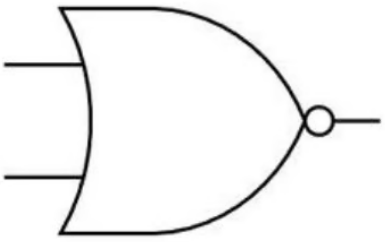
Expression operator	Circuit symbol	Function
(A **NAND** B)		Returns TRUE if **either or both** inputs are FALSE **TRUE** OR **FALSE** = **TRUE** **FALSE** OR **FALSE** = **TRUE** **TRUE** OR **TRUE** = **FALSE**

NOR (not OR)

Expression operator	Circuit symbol	Function

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

(A **NOR** B)		Returns TRUE if **both** inputs are FALSE **TRUE** OR **FALSE** = **FALSE** **FALSE** OR **FALSE** = **TRUE** **TRUE** OR **TRUE** = **FALSE**

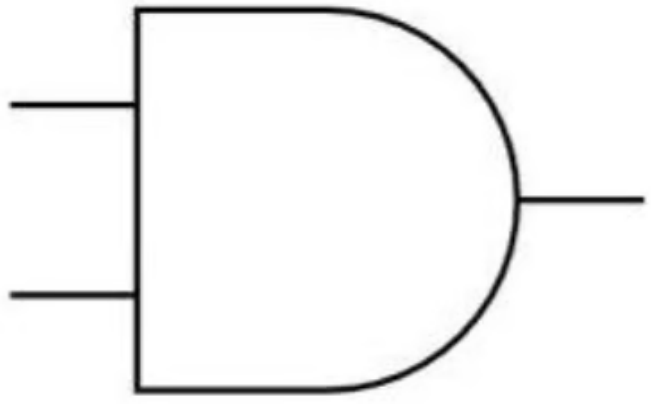
Your notes

Truth tables

What is a truth table?

- A truth table is a **tool** used in logic and computer science to **visualise** the **results** of **Boolean expressions**
- They represent **all possible inputs** and the **associated outputs** for a **given Boolean expression**

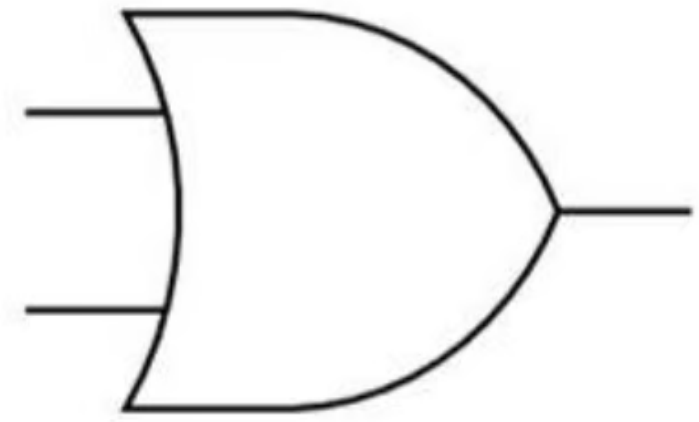
AND

Circuit symbol	Truth Table
	<table><tr><th>A</th><th>B</th><th>A AND B</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>

OR

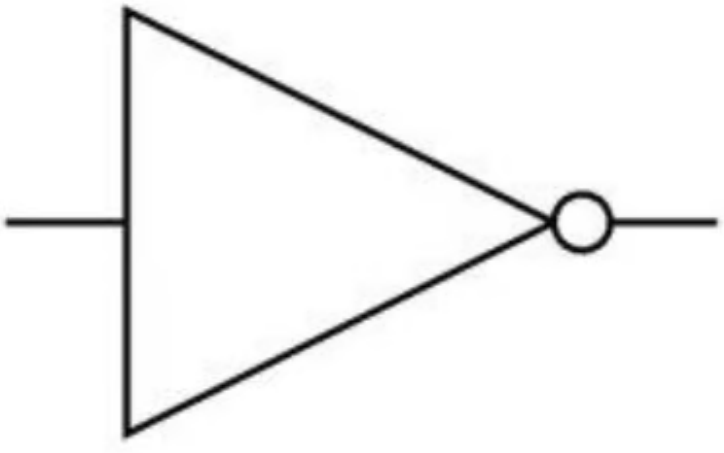
© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Circuit symbol	Truth Table
	A | B | A OR B 0 | 0 | 0 0 | 1 | 1 1 | 0 | 1 1 | 1 | 1

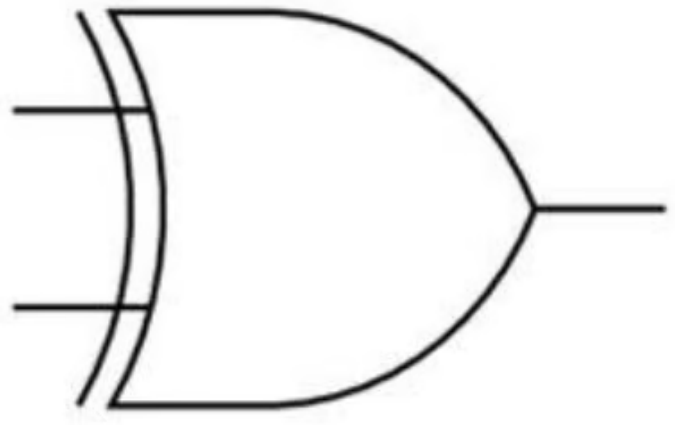
Your notes

NOT

Circuit symbol	Truth Table
	A | NOT 0 | 1 1 | 0

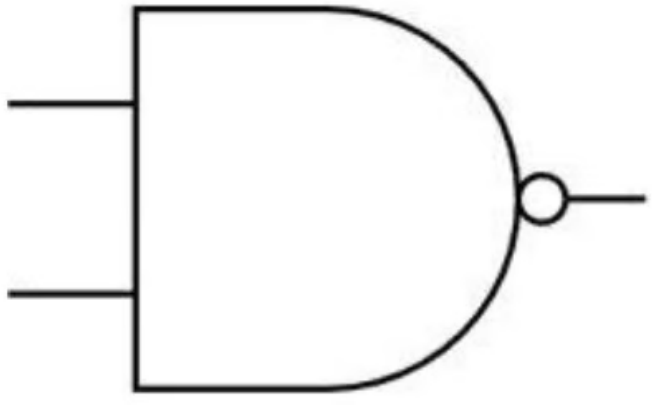
XOR (exclusive)

Study copy distributed by EduResource ZA | eduresourcza.com

Circuit symbol	Truth Table
	A | B | A XOR B 0 | 0 | 0 0 | 1 | 1 1 | 0 | 1 1 | 1 | 0

Your notes

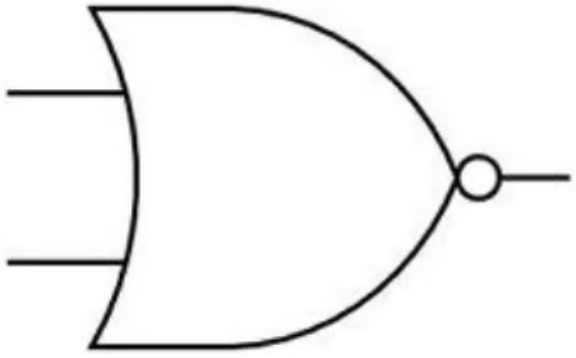
NAND (not and)

Circuit symbol	Truth Table
	A | B | NOT (A AND B) 0 | 0 | 1 0 | 1 | 1 1 | 0 | 1 1 | 1 | 0

NOR (not or)

Circuit symbol	Truth Table

Study copy distributed by EduResource ZA | eduresourcza.com



A	B	NOT (A OR B)
0	0	1
0	1	0
1	0	0
1	1	0

Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

From problem statements

- A problem statement is typically a **written interpretation of a scenario** that requires a **specific logical outcome**
- Logic circuits can be **constructed from problem statements**

Worked Example

A server room has an **automatic cooling fan (F)**
The fan turns on based on the following input parameters:

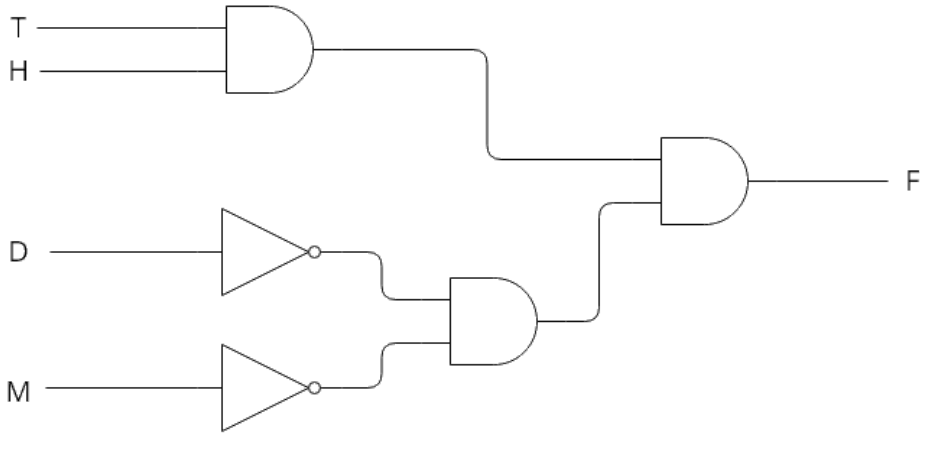
Parameter	Description	Binary Value	Condition
T	Temperature	1 = Too high	0 = Acceptable
H	Humidity	1 = Too high	0 = Acceptable
M	Maintenance mode	1 = Active	0 = Inactive
D	Door sensor	1 = Door open	0 = Door closed

The **fan (F = 1)** turns on if:

- The **temperature is too high**
- **AND** humidity is too high
- **AND** the **door is closed**
- **UNLESS** maintenance mode is active (if maintenance is active, the fan must stay off)

Draw a logic circuit to represent how the **fan (F)** operates based on the input conditions. **[3]**

Answer



© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- **NOT gates** on **D** and **M** inputs [1 mark]
- AND gate combines **T AND H** or **NOT D AND NOT M** [1 mark]
- Combines two previous AND outputs (e.g. (**T AND H) AND (NOT D AND NOT M)**) [1 mark]

Your notes

From logic expressions

- A logic expression is a way of showing how a **logic circuit works using symbols** and **Boolean logic** (AND, OR, NOT)
- It describes the conditions that must be met for the **output** to be **true (1)** or **false (0)**
- From a logic expression, a **logic circuit and/or truth table can be constructed**

Worked Example

A logic expression is given:

S = (A AND B AND C) OR (B XOR C)

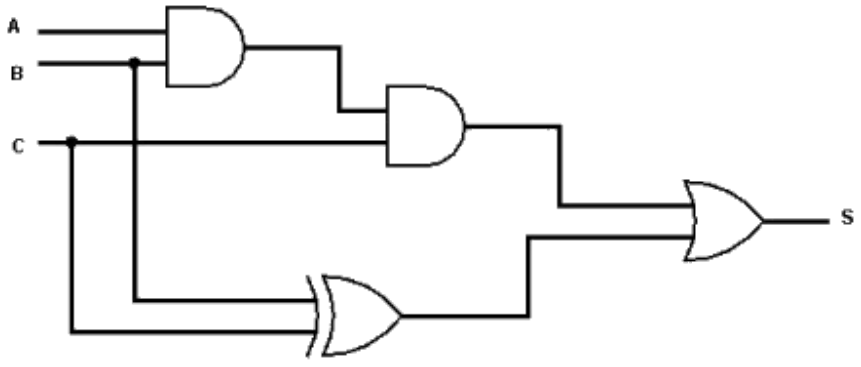
(a) Draw the logic circuit for the given expression **[4]**

(b) Complete the truth table for the logic expression: **[2]**

S = (A AND B AND C) OR (B XOR C)

Answer

(a)



- Each correct gate [1 mark]

(b)

A	B	C	S
0	0	0	0
0	0	1	1

Study copy distributed by EduResource ZA | eduresourcza.com

0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

- Rows **1 to 4** correct [1 mark]
- Rows **5 to 8** correct [1 mark]



Your notes

From truth tables

- To create a truth table for the expression **P = (A AND B) AND NOT C**
  - Calculate the numbers of rows needed ($2^{\text{number of inputs}}$)
  - In this example there are 3 inputs (**A, B, C**) so a total of **8 rows** are needed (2^3)
  - To not miss any combination of inputs, start with **000** and **count** up in **3-bit binary (0-7)**

A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

- Add a new column to show the **results** of the brackets first (A AND B)

A	B	C	A AND B

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com



Your notes

0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**0**
1	0	0	**0**
1	0	1	**0**
1	1	0	**1**
1	1	1	**1**

- Add a new column to show the **results** of NOT C

A	B	C	A AND B	NOT C
0	0	0	0	**1**
0	0	1	0	**0**
0	1	0	0	**1**
0	1	1	0	**0**
1	0	0	0	**1**
1	0	1	0	**0**
1	1	0	1	**1**
1	1	1	1	**0**

- The last column shows the **result** of the Boolean expression (**P**) by comparing (A AND B) AND NOT C //

A	B	C	A AND B	NOT C	P
0	0	0	0	1	**0**

Study copy distributed by EduResource ZA | eduresourcza.com

0	0	1	0	0	**0**
0	1	0	0	1	**0**
0	1	1	0	0	**0**
1	0	0	0	1	**0**
1	0	1	0	0	**0**
1	1	0	1	1	**1**
1	1	1	1	0	**0**

Your notes

Examiner Tips and Tricks

It is possible to create a truth table when combining expressions that show only the inputs and the final outputs.

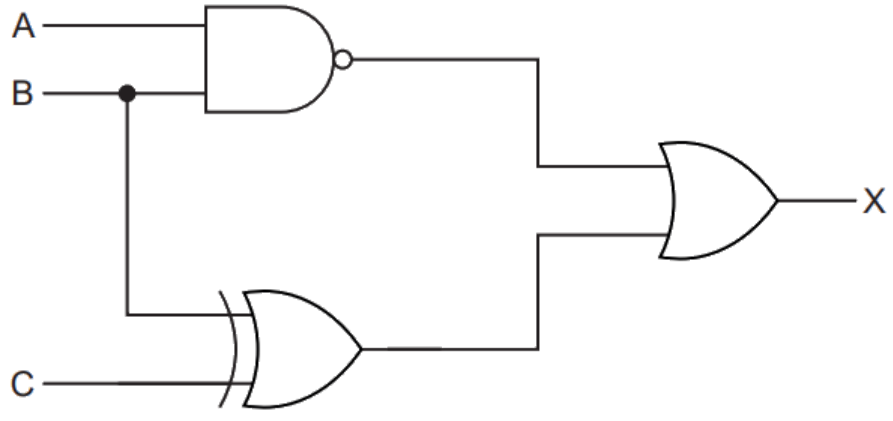
The inclusion of the extra columns **supports the process** but can be skipped if you feel able to do those in your head as you go.

From logic circuits

- From a logic circuit you can create the **logic expression and/or truth table**

Worked Example

Write the logic expression for the given logic circuit. **[3]**



© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Answer

(A NAND B) OR (B XOR C)

- A NAND B [1 mark]
- B XOR C [1 mark]
- OR [1 mark]



Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Central Processing Unit (CPU) Architecture

Contents

- System architecture
- Components of the CPU
- System performance
- Ports & interfaces
- Fetch-Execute Cycle
- Interrupts

© 2026 Save My Exams, Ltd.

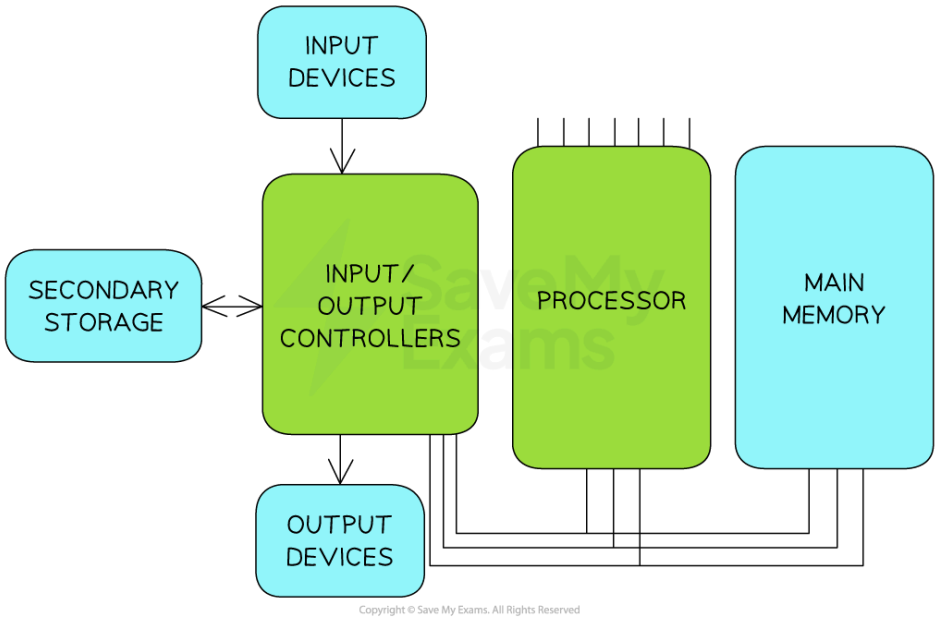
Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Von Neumann Model

What is the Von Neumann model?

- The Von Neumann model is a **design of the CPU** which was proposed by Mathematician John Von Neumann in the 1940s, which most **general-purpose** computers are built upon
- The Von Neumann model outlines **how the computer memory, input/output devices and processor** all work together



The Von-Neumann model

- The Von Neumann model consists of:
  - **Central processing unit (CPU)**
  - **Able to access memory directly**
  - **Memory that can store programs as well as data (registers)**
  - **Stored programs that contain instructions that can be executed in order**

Registers

- Extremely **small**, extremely **fast memory** located **in the CPU**
- Hold small amounts of data **needed as part of the fetch-execute cycle**
- Includes both:
  - **General purpose registers**

Study copy distributed by EduResource ZA | eduresourcza.com

- **Special purpose registers**.
- A **general purpose** register can store **any data the CPU is currently working on**
  - It is **flexible** and can be used for various operations



Your notes

Special purpose registers

What is a special purpose register?

- A special purpose register **has a dedicated role** within the operation of the CPU
- There are a number of special purpose registers that **control or track data**
- Examples of special purpose registers includes:
  - **The Program Counter (PC)**
  - **The Memory Address Register (MAR)**
  - **The Memory Data Register (MDR)**
  - **The Accumulator (ACC)**
  - **Current Instruction Register (CIR)**
  - **Index register (IX)**
  - **Status register (SR)**
- For each of the registers you must know
  - **The name of the register**
  - **Its acronym**
  - **The purpose of the register**

Name	Acronym	Purpose
Program Counter	PC	▪ **Holds the memory address** of the **next instructions** to be executed ▪ **Increments by 1** as the fetch-decode-execute cycle runs
Memory Address Register	MAR	▪ **Holds the memory address** of where data or instructions are **to be fetched from memory**
Memory Data Register	MDR	▪ **Stores the data or instruction** which has been fetched from memory
Current Instruction Register	CIR	▪ **Stores the instruction** the CPU is currently decoding or executing

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Accumulator	**ACC**	▪ **Stores the results** of any calculations that have taken place in the Arithmetic Logic Unit (ALU)
Index Register	**IX**	▪ **Stores a value** that can be **added to an address** to get the effective memory address ▪ Used for **indexed addressing**
Status Register	**SR**	▪ **Stores flags** that reflect the **outcome of the CPU operations** ▪ Holds **individual bits (flags)** which are **set or cleared depending on the result of an instruction** (e.g. overflow)



Your notes

© 2026 Save My Exams, Ltd.

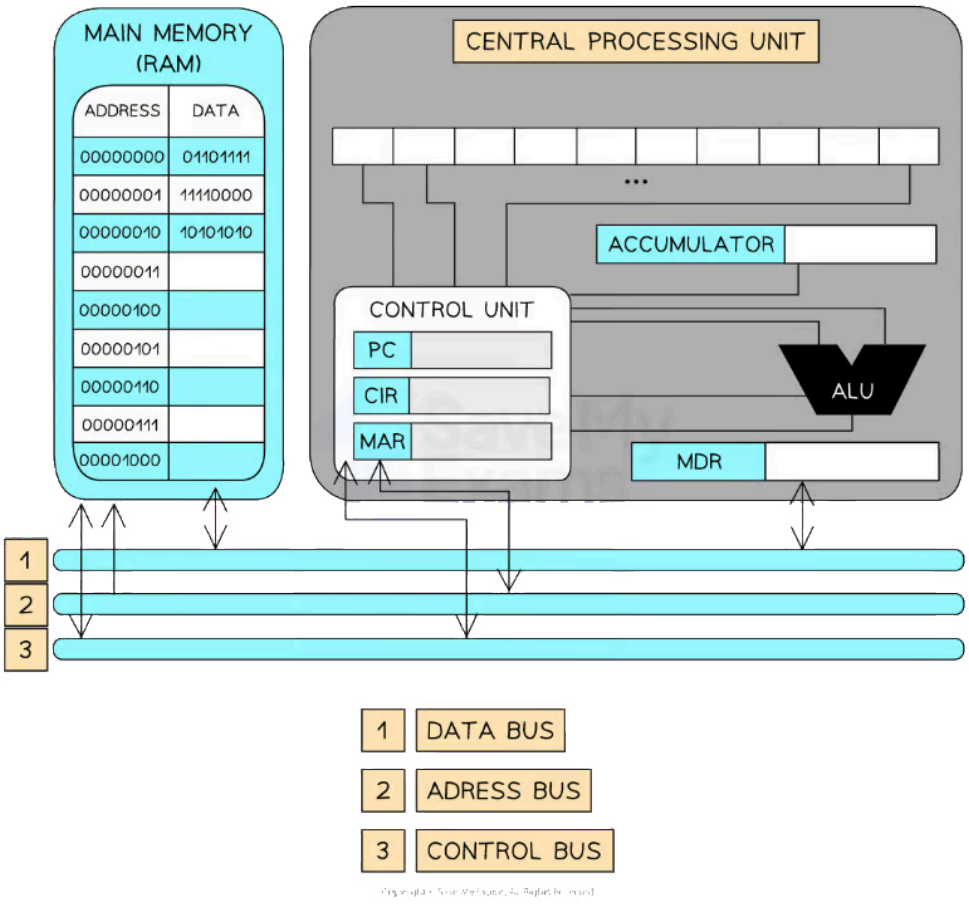
Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Components of the CPU

What is the CPU?

- The **CPU** is responsible for **processing all data** within the computer
- It is made up of a number of **components** including:
  - **Arithmetic and Logic Unit (ALU)**
  - **Control Unit (CU)**
  - **System clock**
  - **Immediate Access Store (IAS)**
  - **Buses**



Component	Full name	Purpose / function

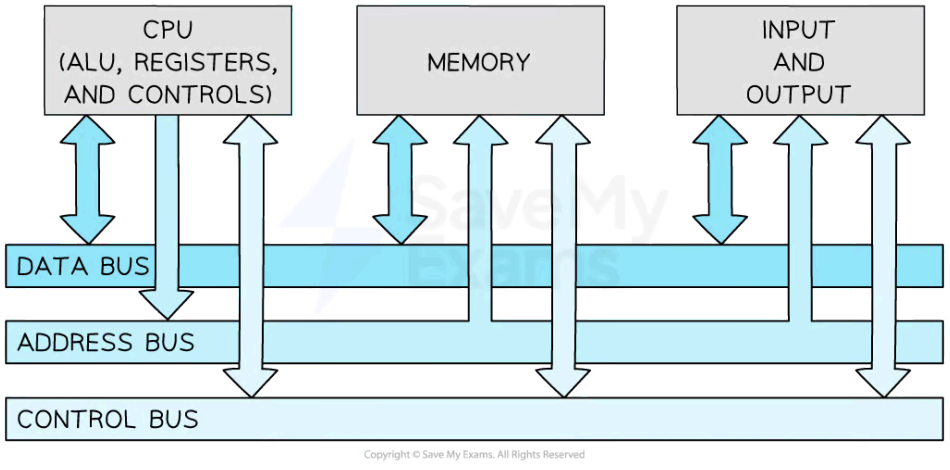
Study copy distributed by EduResource ZA | eduresourcza.com

ALU	Arithmetic Logic Unit	Performs **calculations** (e.g. add, subtract) and **logic operations** (e.g. AND, OR)
CU	Control Unit	**Manages and controls** the execution of instructions and **directs data flow**
System Clock	–	Sends out **regular pulses** to **synchronise** all CPU operations
IAS	Immediate Access Store	**Main memory** where **instructions and data** needed by the CPU are **temporarily stored**

Your notes

Buses

- Components within the CPU and wider computer system are **connected by buses**
- A bus is **a set of parallel wires** through which **data/signals are transmitted** from one component to another
- There are 3 types of bus:
  - **Address** - **unidirectional**, carries **location data** (addresses), data is written to/read from
  - **Data** - **bidirectional**, carries **data or instructions**
  - **Control** - bidirectional, carries **commands** and **control signals** to tell components when they should be receiving reads or writes etc..



Copyright © Save My Exams. All Rights Reserved

Study copy distributed by EduResource ZA | eduresourcza.com

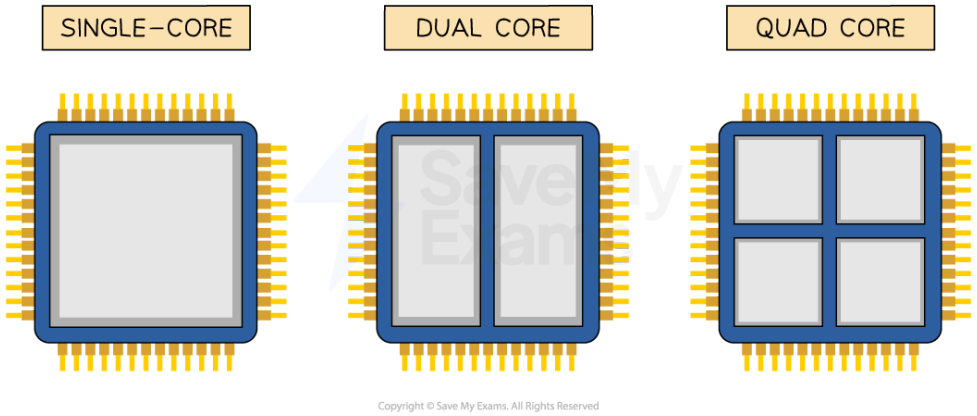
Your notes

Cores

What is a core?

- A core acts like a **mini CPU**, it can **fetch, decode, and execute** instructions on its own
- Each core has its own:
  - **Control Unit (CU)**
  - **Arithmetic Logic Unit (ALU)**
  - **Registers**
- A CPU with **more than one core** is a **multicore CPU**
- Allows for **parallel processing**, multiple instructions are processed **at the same time**
- Example:
  - A **dual-core** processor has **2 cores**
  - A **quad-core** processor has **4 cores**
- More cores = **better performance**, especially for powerful programs

  Example:
  - A **quad-core CPU** running at **3GHz**
  - 4 cores × 3 billion = **12 billion instructions per second**
- A **dual-core** processor isn't always **twice as fast** as a single-core
  - Some time is used for **organising tasks between cores**
- Not all tasks can be **split across cores**
  - Some tasks are **sequential** and must be done **step-by-step**



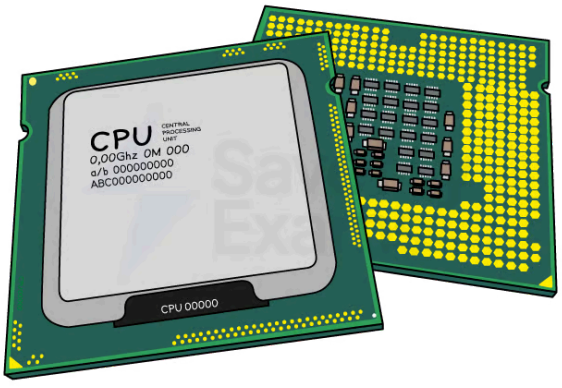
Copyright © Save My Exams. All Rights Reserved

Study copy distributed by EduResource ZA | eduresourcza.com

Clock speed

What is clock speed?

- The **clock** controls the **timing of operations** inside the CPU
- It constantly **switches between 0 and 1**, each switch is called a **state change**
- A **state change** can represent **one step** in the **fetch-decode-execute cycle**
  - Some instructions may take **more than one cycle**
- **Clock speed** measures **how many state changes happen per second**
- **1 cycle per second = 1 Hz**
- A typical clock speed is around **2.3 GHz**
  - That's **2.3 billion cycles per second**
- A **higher clock speed** means the CPU can **execute more instructions per second**
- This helps the computer **run tasks more quickly and efficiently**



Copyright © Save My Exams. All Rights Reserved

Your notes

Cache memory

What is cache memory?

- Cache is part of **primary storage**
- Stores **frequently used data and instructions**
- Located **closer to the CPU** than RAM, so it's **faster to access**
- Some cache is built **directly into each processor core**
- **Speeds up** the performance of the CPU

Study copy distributed by EduResource ZA | eduresourcza.com

- More cache = **more data stored** nearby = **less time waiting** for data from RAM
- Example:
  - A **website** you visit often can be stored in the cache
  - Next time you visit, it **loads faster**
  - If the website updates, the **cached version is updated too**

Your notes

Levels of cache

Level	Location	Speed	Size
Level 1 (L1)	Inside each CPU core	**Fastest**	**Smallest**
Level 2 (L2)	Inside or near each core	Fast	Medium
Level 3 (L3)	Shared by all cores	Slower than L1/L2	**Largest**

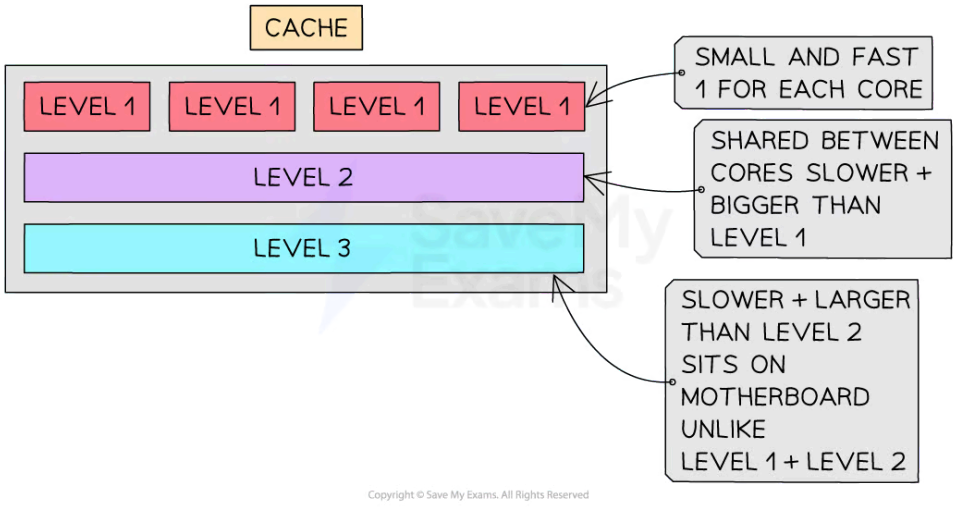


Diagram of the levels within the cache

Bus width

What is bus width?

- Bus width is the number of **bits** a bus can carry at once
- A **wider bus** can transfer **more bits in a single operation**
- Example:
  - A **32-bit bus** transfers 32 bits at a time
  - A **64-bit bus** transfers 64 bits at a time

Study copy distributed by EduResource ZA | eduresourcza.com

- **Increases performance**, more data can be moved in the same amount of time
- A **wider data bus** means:
  - **Faster processing**
  - More **efficient** memory access
- Important for **high-performance tasks** like gaming, video editing, or large data processing



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

USB

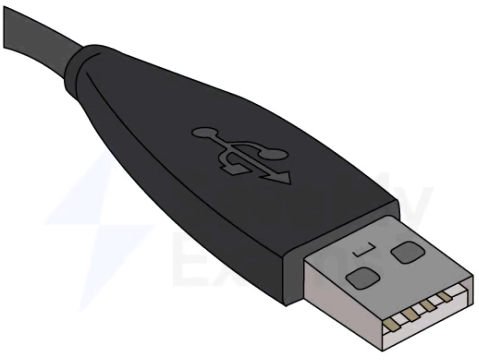
What is USB?

- The **Universal Serial Bus** (**USB**) is a widely used **standard for transmitting data** between devices
- It is a **serial communication method**, and it **operates asynchronously**
- Many devices use USB such as:
  - **Keyboards**
  - **Mice**
  - **Video cameras**
  - **Printers**
  - **Portable media players**
  - **Mobile phone**
  - **Disk drives**
  - **Network adapters**
- Different USB **connector types** exist for different devices
- The letters refer to the **physical shape and design** of the USB connector:
  - **USB-A** - Commonly used for flash drives, mice, keyboards, external HDD
  - **USB-B** - Found in printers, scanners, and older external storage devices
  - **USB-C** - Latest standard, known for it's **small size**, **transfer speeds**, and it's **ability to carry power**
- The term USB can also be **followed by numbers** (**USB 2.0, 3.0, 4 etc**.)
- The numbers refer to the **generation of USB technology**, which determines the **speed and performance**:
  - **USB 1.1** - 12 Mbps (very slow)
  - **USB 2.0** - 480 Mbps (very common but slower compared to modern versions)
  - **USB 3.0/3.1/3.2** - 5 Gbps to 20 Gbps (much faster, used for external HDDs and gaming devices)
  - **USB4/ USB4 2.0** - Up to 80 Gbps (the latest and fastest, used for high speed data transfer)
- When a device is connected to a USB port the computer:

© 2026 Save My Exams, Ltd.

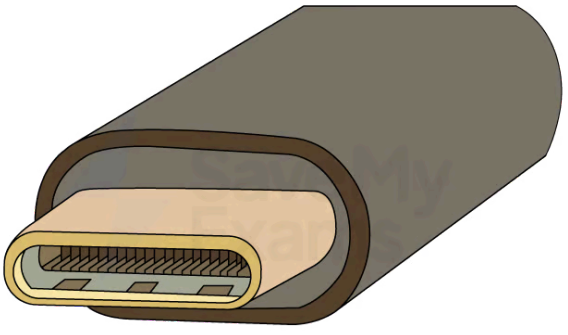
Study copy distributed by EduResource ZA | eduresourcza.com

- **Automatically detects** that the device has been **connected**
- **Looks for the correct driver:**
  - If the driver is already installed, the **appropriate device driver is loaded** so that the **device can communicate with the computer**
  - If the device is **new**, the computer will **look for a compatible device driver**
  - If one cannot be found, the **user must download and install** an appropriate driver **manually**



A USB–A CONNECTOR

Copyright © Save My Exams. All Rights Reserved



A USB–C CONNECTOR

Copyright © Save My Exams. All Rights Reserved

Advantages and disadvantages of USB

Advantages	Disadvantages
Devices are **automatically detected** and **drivers are automatically loaded** for communication	The **maximum** cable length is **roughly 5 metres** meaning it cannot be used over long distances, **limiting its use**
Cable connectors fit in only one way.	Older versions of USB have **limited transmission rates** for example **USB**

Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

This prevents incorrect connections and **ensures compatible data transmission**	**2.0** has **480Mbps**
As USB usage is standardised, there is a lot of support available online and from retailers	**Very old USB standards may not be supported** in the near future (USB 1.1, USB 2.0, etc)
Several different **data transmission rates** are supported. The newest transmission rate as of 2022 is **USB4 2.0** with **80 Gbps** (81,920 Mbps, 170× faster than USB 2.0)	
Newer USB standards are **backwards compatible** with older USB standards	



Your notes

HDMI

What is HDMI?

- HDMI is a **digital port** that sends both **video and audio output** from a computer to HDMI-enabled devices (e.g. TVs, monitors)
- Replaces older **VGA** analogue systems
- Modern HD TVs and displays require more data because they support:
  - **Widescreen format** (16:9 aspect ratio)
  - **Higher resolution** (e.g. 1920 × 1080 pixels)
  - **Faster refresh rates** (e.g. 120 Hz)
  - **Wider colour range** (millions of colour variations)
- These features need **faster and higher bandwidth**, which **HDMI provides** (up to **10 Gbps**)
- HDMI uses **HDCP (High-bandwidth Digital Content Protection)**
- Devices check for an **authentication key** before sending data (e.g. Blu-ray player to HD TV)
- If the device is authenticated, a **"handshake"** occurs and data is transmitted
- Helps prevent **unauthorised copying** of protected content

VGA

What is VGA?

- VGA was Introduced in the late 1980s as the **standard for video output**
- It is now considered **outdated** and is being **phased out**

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- Maximum resolution: **640 × 480 pixels**
- Refresh rate: **up to 60 Hz**, but only supports **16 colours** at that rate
- If resolution is reduced (e.g. **320 × 200**), it can support up to **256 colours**
- VGA is an **analogue signal**, which means:
  - **Lower image quality over long cables**
  - **More likely to experience signal loss or interference**



Your notes

© 2026 Save My Exams, Ltd.

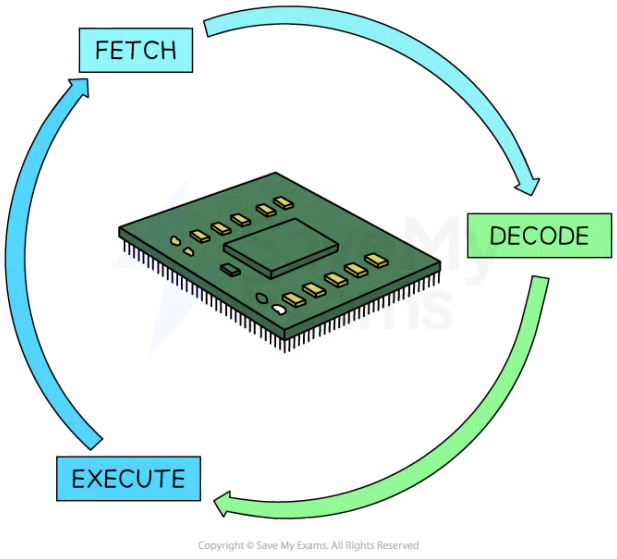
Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

F-E stages

What is the fetch-execute cycle?

- The fetch-execute cycle is the **process** that the CPU goes through **repeatedly to process instructions**
- There are **3 stages**:
  - **Fetching** an instruction from memory - supplying the address and receiving the instruction from memory
  - **Decoding** the instruction - interpreting the instruction and then reading and retrieving the required data from their addresses
  - **Executing** the instruction - the CPU carries out the required action



Copyright © Save My Exams. All Rights Reserved

How are registers used in the fetch-execute cycle?

- In the section below, registers and CPU components appear in **bold** and assembly language is in *italics*
- During the fetch-execute cycle, the following steps happen:
- **Fetch**
  - The **PC** is loaded with 0
  - The value from the **PC** (0) is copied to the **MAR**
  - The address in the **MAR** is sent via the **address bus**, and a **read** instruction is sent on the **control bus**

Study copy distributed by EduResource ZA | eduresourcza.com

- The data from that memory location is sent down the **data bus** to the **MDR**
- The **PC** is incremented by 1, ready for the next instruction

- **Decode**
  - The data from the **MDR** is copied to the **CIR**
  - The **CIR** splits the instruction into **opcode** and **operand**
  - The instruction is sent to the **CU**, which decodes what to do next
- **Execute**
  - The registers used will depend on the instruction:
    - *INP* – value from input device is stored in the **ACC**
    - *OUT* – value currently in the **ACC** is sent to the output
    - *LDA* – loads value from RAM into the **MDR**, then to the **ACC**
    - *STA* – takes the value from the **ACC**, moves it to the **MDR**, then stores it in RAM (location in **MAR**)
    - *ADD/SUB* – value in RAM is loaded into the **MDR**, then the calculation happens in the **ALU**, result stored in the **ACC**
    - *BRA/BRZ/BRP* – branching instructions are handled by the **ALU** to check if conditions are met, and the **PC** is updated accordingly



Your notes

Register Transfer Notation (RTN)

- Register Transfer Notation (RTN) is a way of describing how **data moves between registers** in a CPU during the **fetch–execute cycle**
- It uses **symbols** to represent **registers**, and the arrow symbol ← to show **data being transferred** from one place to another
- It is a shorthand way to show what's happening inside the CPU

Symbol	Meaning
←	Data is transferred into a register
PC	Program Counter
MAR	Memory Address Register
MDR	Memory Data Register
CIR	Current Instruction Register
ACC	Accumulator

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

RAM[ ]	Memory location (RAM[address])



Your notes

Fetch

MAR ← PC ; Copy address from Program Counter to MAR

MDR ← RAM[MAR] ; Read instruction from memory into MDR

PC ← PC + 1 ; Increment PC for the next instruction

CIR ← MDR ; Copy instruction from MDR to CIR

Decode

- No specific register transfer, the **CU** decodes the instruction in the **CIR**

Execute examples

- **LDA X** (Load value from memory location **X** into **ACC**):

MAR ← operand ; Load address X into MAR

MDR ← RAM[MAR] ; Load data from RAM into MDR

ACC ← MDR ; Copy data into ACC

- **STA X** (Store value from **ACC** into memory location **X**):

MAR ← operand ; Load address X into MAR

MDR ← ACC ; Copy data from ACC to MDR

RAM[MAR] ← MDR ; Store MDR value into memory

- **ADD X** (Add value from memory location **X** to **ACC**):

MAR ← operand

MDR ← RAM[MAR]

ACC ← ACC + MDR

- **BRZ X** (Branch to **X** if **ACC** = 0):

If ACC = 0 then PC ← operand

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Interrupt handling

What are interrupts?

- An interrupt is a **signal** to the processor that **stops its current task** and performs a different task temporarily
- Interrupts can be **hardware events or time-sensitive tasks**
- When an interrupt occurs, the processor **suspends the current program** execution and transfers control to an **interrupt service routine**

Purpose and role of interrupts

- **Real-time Event Handling**: hardware errors and signals from input devices e.g. hard disk failure
- **Device Communication**: alerts from external devices e.g. printer jams and network errors
- **Multitasking**: suspending processing in one application so that the user can switch to another

Types of interrupts

Type	Definition	Example
Hardware Interrupts	Generated by external devices	Keyboard input, mouse movements, disk I/O requests
Software Interrupts	Triggered by software or the operating system	Application requests to open a file, **division by zero** errors
Trap Interrupts	Intentionally triggered by a program	Software debugging, handling unexpected error cases

The interrupt process

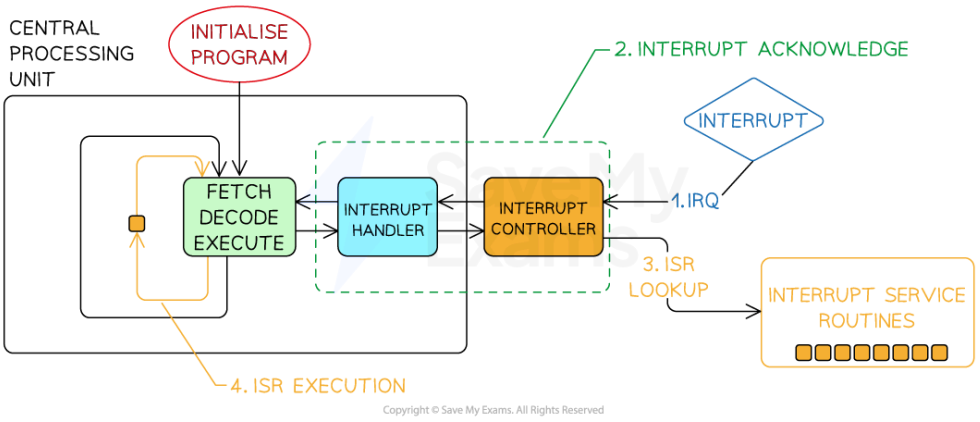
1. **Interrupt Request (IRQ)**
    - An external device or software generates an interrupt, signalling the processor to stop its current task
    - The interrupt controller passes this to the interrupt handler for assessment
2. **Interrupt acknowledge**
    - The interrupt handler decides if the interrupt needs to be dealt with n**ow or later**

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- If yes, the **current contents of the processor registers are saved** in memory

3. **Interrupt Service Routine (ISR) lookup**
    - The processor **fetches the corresponding Interrupt Service Routine (ISR)** associated with the interrupt type
4. **ISR execution**
    - The processor **transfers control** to the ISR and **executes the routine** to handle the specific interrupt
5. **Interrupt exit**
    - After the ISR completes, **the processor restores the content of the registers** from step 2
    - The fetch-decode-execute cycle is **resumed**



The interrupt process

What is an ISR?

- An ISR is a **special function** that handles a particular interrupt type
- Each type of interrupt has a **corresponding routine**
    - E.g. printer jam, hard disk failure, file download error, network connection error all have routines to be followed when they happen
- ISRs should be **concise**, **efficient**, and carefully designed to **minimise the time taken to execute**, as they often need to handle time-sensitive events

Interrupt priority and nesting

- Interrupt prioritisation means the processor can acknowledge and **switch to resolving a higher-priority interrupt**
- Prioritising interrupts is vital because many things can go wrong **at the same time**
- **Lower-priority ISRs may be temporarily suspended** until the higher-priority ISR completes the execution

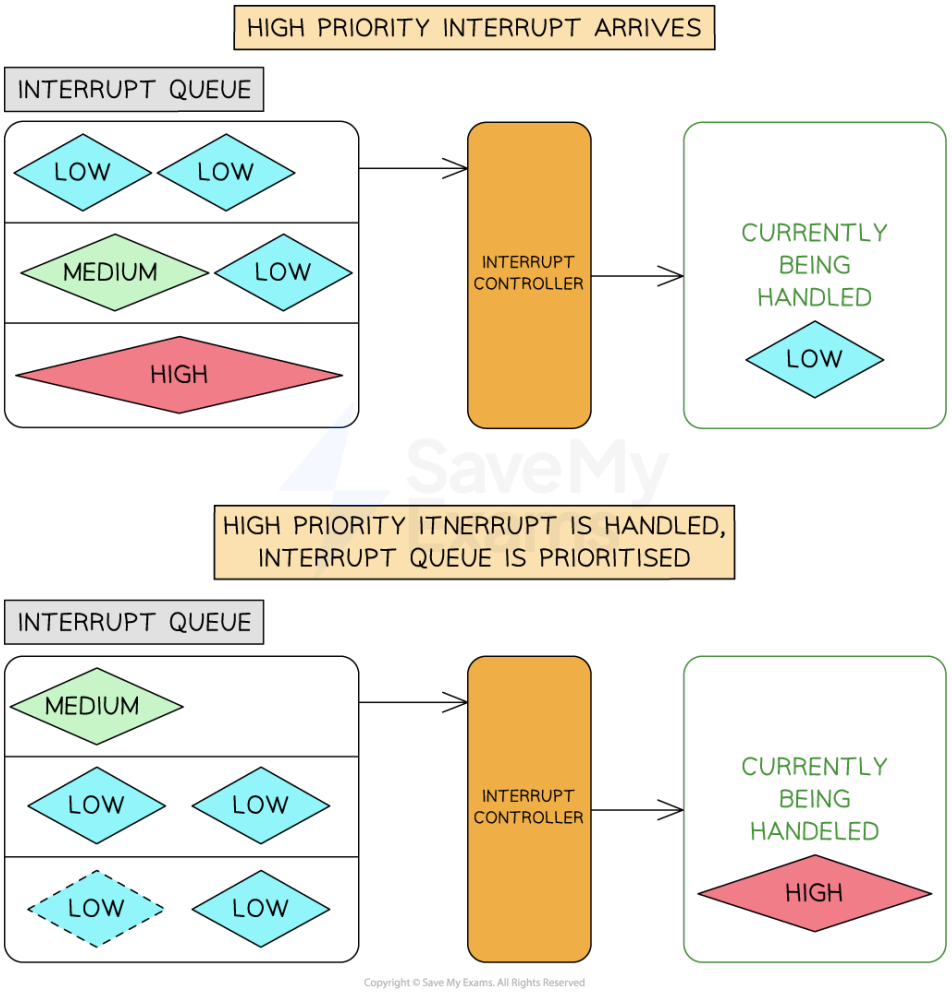
Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

- **Nesting of interrupts** refers to the ability of the processor to handle interrupts within interrupts
- Proper management of nested interrupts **avoids potential conflicts** and ensures system stability

Nesting of interrupts

Interrupt priority handling



System handling of interrupt priority

Worked Example

Explain how the computer handles an interrupt when a key is pressed on the keyboard. **[5]**

Answer

- An interrupt flag is raised in the interrupt register [1 mark]
- The system finishes its current fetch-execute cycle [1 mark]

Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

- The interrupt register is checked for a higher priority interrupt [1 mark]
- If a higher priority interrupt is found, the current register contents are pushed to the stack [1 mark]
- The appropriate Interrupt Service Routine (ISR) is executed [1 mark]
- The ISR processes the key press and completes [1 mark]
- The register contents are restored from the stack [1 mark]
- Control is returned to the original process [1 mark]



Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Assembly Language

Contents

- Assembly language basics
- Assembly process
- Instruction groups
- Addressing modes

© 2026 Save My Exams, Ltd.

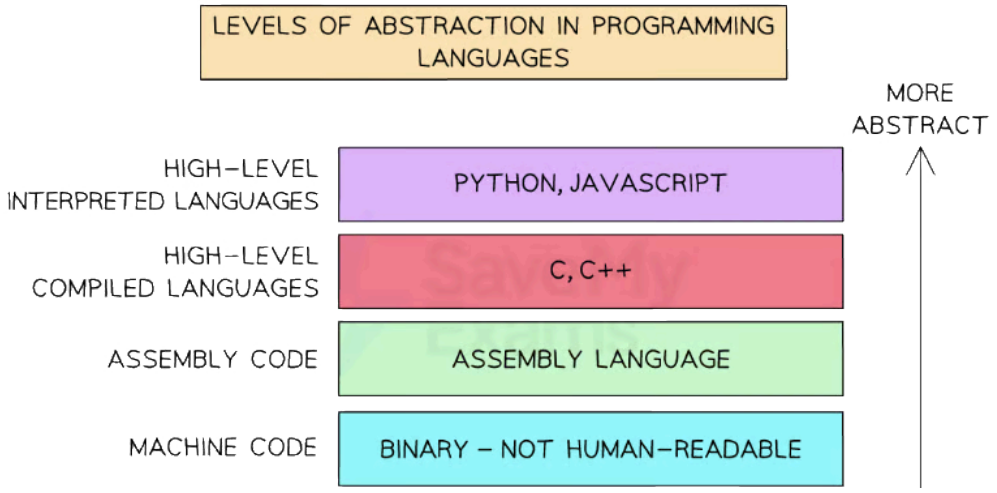
Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Machine code vs assembly

What is machine code?

- **Machine code** is a first-generation language
- Instructions are **directly executable by the processor**
- Written in **binary code**



Levels of Abstraction of Programming Languages

What is assembly?

- **Assembly code** is a second-generation language
- The code is written using **mnemonics**, abbreviated text commands such as **LDA (Load), STA(Store)**
- Using this language programmers can write **human-readable programs** that correspond **almost exactly to machine code**
- **One assembly language instruction** translates to **one machine code instruction**
- **Needs to be translated** into machine code for the computer to be able to execute it
- Each type of computer CPU has a **specific instruction set**

Instruction set

- An instruction set is a list of **all the commands** that can be **processed by a CPU**
- Each command has a **binary code** which is the **machine code**
- In assembly the binary code is written using mnemonics and split into an **opcode** and **operand**

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- The **opcode** is the **part of an instruction** that tells the CPU what **operation** to perform
- It's short for **"operation code"**
- Example operations: *LDA* (load), *STA* (store), *ADD*, *INP*, *OUT*, *BRZ*, *BRA*
- The **operand** is the **data** or **memory address** the opcode will work with
- It gives the **extra detail** the CPU needs to complete the instruction
- Often used to specify a **memory address**, a **value**, or a **label**



Your notes

Opcode	Operand	Explanation
LDM	#n	Immediate addressing. Load the number n to the accumulator (ACC).
LDD	<address>	Direct addressing. Load the contents of the location at the given address to ACC.
LDI	<address>	Indirect addressing. Use the value at the given address as a new address. Load contents to ACC.
LDX	<address>	Indexed addressing. Use <address> + index register to get the final address. Load to ACC.
LDR	#n	Immediate addressing. Load the number n to the index register (IX).
MOV	<register>	Move the contents of ACC to the given register (e.g. IX).
STO	<address>	Store the contents of ACC at the given memory address.
ADD	<address>	Add the contents of the given address to ACC.
SUB	<address>	Subtract the contents of the given address from ACC.
INC	<register>	Increment the contents of the register (ACC or IX) by 1.
DEC	<register>	Decrement the contents of the register (ACC or IX) by 1.
JMP	<address>	Unconditional jump to the given address.
CMP	<address>	Compare the contents of ACC with the contents of the given address.
CMP	#n	Compare the contents of ACC with the number n.

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

CMI	<address>	Indirect addressing. Compare ACC with the contents at the address stored in <address>.
JPE	<address>	Jump to <address> if the previous compare result was **true** (equal).
JPN	<address>	Jump to <address> if the previous compare result was **false** (not equal).
IN		Take input from the keyboard and store its ASCII value in ACC.
OUT		Output to the screen the character stored in ACC.
END		End the program and return control to the operating system

All questions will assume there is only one general purpose register available (Accumulator)

ACC denotes Accumulator

IX denotes Index Register

<address> can be an absolute or symbolic address

denotes a denary number, e.g. #123

B denotes a binary number, e.g. B01001010

& denotes a hexadecimal number, e.g. &4A



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Two-pass assembler

What is a two-pass assembler?

- A two-pass assembler **translates assembly language into machine code** in two stages
- It produces an **object program** that can be **stored**, **loaded**, and **run later**
- To execute this program, another utility called a **loader** is needed
- The assembler scans the source code **twice**:
  - In the **first pass**, it identifies all **labels** and records their memory addresses
  - In the **second pass**, it replaces those labels with the correct **memory addresses** in the machine code
- This **ensures the final machine code is accurate** and ready to be executed

Example

- The following program loads the value of A (5), adds the value of B (3), stores the result (8) in C, then halts:

```
    LOAD A
    ADD B
    STORE C
    HALT
A:  DATA 5
B:  DATA 3
C:  DATA 0
```

- **Pass 1:**
  - The assembler reads each line and assigns memory addresses to labels (e.g. A, B, C), but **does not translate the instructions yet**

Address	Instruction	Label	Add to symbol table
00	LOAD A		
01	ADD B		
02	STORE C		
03	HALT		
04	DATA 5	A	**A** = 04

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

05	DATA 3	B	**B** = 05
06	DATA 0	C	**C** = 06



Your notes

Symbol table created:

Label	Address
A	04
B	05
C	06

- **Pass 2:**
  - The assembler replaces labels with their corresponding memory addresses and outputs machine code
  - In this example we assume the following opcodes for simplicity:

Mnemonic	Opcode
LOAD	01
ADD	02
STORE	03
HALT	00
DATA	-

- Now generate the object (machine) code:

Address	Assembly	Machine code	Explanation
00	LOAD A	01 04	LOAD from address 04
01	ADD B	02 05	ADD value at address 05
02	STORE C	03 06	STORE into address 06

Study copy distributed by EduResource ZA | eduresourcza.com

03	HALT	00 00	Stop program
04	DATA 5	00 05	Data value 5
05	DATA 3	00 03	Data value 3
06	DATA 0	00 00	Data value 0 (placeholder)

Final object program:

```
01 04
02 05
03 06
00 00
00 05
00 03
00 00
```

Program tracing

What is a trace table?

- A **trace table** is a table used to **manually track the values of variables** as a program runs
- It helps to follow the flow of a program **line by line**, checking whether the logic works as expected
- Trace tables are used to:
  - **Understand how a program works**
  - **Debug errors** in the logic
  - **Predict outputs** before running the program
  - **Check variable values** at each step

Example

- An assembly program that loads the value from A (4) into the accumulator, adds the value from B (2), stores the result (6) into C, and halts

```
    LOAD A
    ADD B
    STORE C
    HALT
A:  DATA 4
B:  DATA 2
C:  DATA 0
```

- To trace the program, assumptions must be made:
  - **Accumulator (ACC)**: Used for calculations
  - Memory is addressed from 00 onwards

Your notes

Study copy distributed by EduResource ZA | eduresourcza.com



Your notes

- Simple instructions:
  - LOAD X → ACC = memory[X]
  - ADD X → ACC = ACC + memory[X]
  - STORE X → memory[X] = ACC
  - HALT → Stop

Trace table

Step	Instruction	ACC (Accumulator)	Memory[A]	Memory[B]	Memory[C]
0	(Start)	0	4	2	0
1	LOAD A	4	4	2	0
2	ADD B	6	4	2	0
3	STORE C	6	4	2	6
4	HALT	6	4	2	6

Final outcome:

- **ACC = 6**
- **C = 6**
- The program has successfully added A and B, then stored the result in C

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Categories of instructions

What are the categories of instructions?

- Assembly language instructions can be grouped into:
  - **Data movement**
  - **Input and output of data**
  - **Arithmetic operations**
  - **Unconditional and conditional instructions**
  - **Compare instructions**
- For each category, a table containing the instruction (Opcode and Operand), and an explanation is given

Category	Opcode	Operand	Explanation
Data Movement	LOAD	A	Loads the value stored at memory location A into the accumulator.
	STORE	B	Stores the value from the accumulator into memory location B.
	MOV	A, B	Copies the value from register/memory A to register/memory B.
	CLEAR	ACC	Clears the accumulator (sets it to 0).
	DATA	5	Declares a constant or data value (e.g. DATA 5 stores the value 5).
Input and Output of Data	IN		Takes input from a user or input device and stores it in the accumulator.
	OUT		Outputs the value in the accumulator to a screen or output device.
Arithmetic Operations	ADD	A	Adds the value at memory location A to the accumulator.

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

	SUB	B	Subtracts the value at memory location B from the accumulator.
	INC		Increments the accumulator by 1.
	DEC		Decrements the accumulator by 1.
Unconditional and Conditional	JMP	LABEL	Unconditionally jumps to the instruction at LABEL.
	JZ	LABEL	Jumps to LABEL if the accumulator is zero.
	JNZ	LABEL	Jumps to LABEL if the accumulator is **not** zero.
	JC	LABEL	Jumps to LABEL if the carry flag is set.
	HLT		Stops the program (halts execution).
Compare Instructions	CMP	A	Compares the accumulator with the value at memory location A.
	TEST	A	Performs a logical AND between accumulator and A; sets flags, no output.



Your notes

Study copy distributed by EduResource ZA | eduresourcza.com



Your notes

Addressing methods

What is an addressing method?

- An addressing method is a way in which **an instruction in assembly language or machine code** can **access data stored in memory**
- There are five main addressing methods:
  - **Immediate**
  - **Direct**
  - **Indirect**
  - **Indexed**
  - **Relative**

Addressing mode	Description	Example instruction	Explanation
Immediate	The **operand is a constant value** included directly in the instruction	MOV AX, 1234h	Moves the **immediate value** 1234h into the AX register
Direct	The **exact memory address** of the operand is given in the instruction	MOV AX, [1234h]	Moves the value **stored at memory location** 1234h into the AX register
Indirect	A **register contains the memory address** of the operand	MOV AX, [BX]	If BX = 2000h, this moves the value **at memory location** 2000h into AX
Indexed	Uses a **base register and an index** to calculate the memory address	MOV AX, [BX + SI]	If BX = 0050h and SI = 1000h, this moves the value **at** 1050h into the AX register
Relative	The operand is an **offset** relative to the **current instruction address**	JMP +5	Tells the program to **jump forward 5 instructions** from the current line Used in branching

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Bit Manipulation

Contents

- Bitwise operations

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Binary shifts

What are binary shifts?

- A binary shift is an operation that **moves all the bits in a binary number left or right** by a certain number of positions
- Often used in programming and computer systems for **fast multiplication or division by powers of 2**, or for manipulating individual bits
- There are three main types of binary shift:
  - **Logical**
  - **Arithmetic**
  - **Cyclic**

Logical

- **Moves bits left or right** and **fills the gap with 0s**
- Used for **unsigned binary numbers** or raw bit manipulation

Direction	What happens
Left	All bits shift left, a 0 fills the rightmost bit
Right	All bits shift right, a 0 fills the leftmost bit

Left

- The following number is shifted by **two** places to the left

Step	128	64	32	16	8	4	2	1
Original	0	0	0	0	1	1	1	0
Left shift by 2	0	0	1	1	1	0	0	0

- **Original number:** 0000 1110 = **14**
- **Left shift (2) result:** 0011 1000 = **56**
- Each left shift has **doubled** the number:
  - **Original value** = **14**

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- **Left shift 1** - Doubled the number to 28
- **Left shift 2** - Doubled the number to 56



Your notes

Right

- The following number is shifted by **three** places to the right

Step	128	64	32	16	8	4	2	1
Original	1	1	0	0	1	0	0	0
Right shift by 3	0	0	0	1	1	0	0	1

- **Original number:** 1100 1000 = **200**
- **Right shift (3) result:** 0001 1001 = **25**
- Each right shift has **halved** the number:
  - **Original value** = **200**
  - **Right shift 1** - Halved the number to 100
  - **Right shift 2** - Halved the number to 50
  - **Right shift 3** - Halved the number to 25

Arithmetic

- **Moves bits**, but **preserves the sign bit** (used for signed binary numbers in two's complement)

Direction	What happens
Left	Same as logical left (shift left, 0 in)
Right	Shift right, **copy the sign bit (MSB)** into the new leftmost bit

Right

- The following number is shifted by **three** places to the right

Step	128	64	32	16	8	4	2	1
Original	1	1	1	0	1	0	0	0
Right shift by 3	1	1	1	1	1	1	0	1

- **Original number**: 1110 1000 = **-24**

© 2026 Save My Exams, Ltd. | Get more and ace your exams at savemyexams.com

Study copy distributed by EduResource ZA | eduresourcza.com

- **Right shift (3) result**: 1111 1101 = **−3**
- Each arithmetic right shift **divides the number by 2**, rounding **towards negative infinity** (preserving the sign bit)
  - **Original value** = **−24**
  - **Right shift 1** → 1111 0100 = **−12**
  - **Right shift 2** → 1111 1010 = **−6**
  - **Right shift 3** → 1111 1101 = **−3**



Your notes

Cyclic

- **Rotates** the bits around, nothing is lost, no 0s added
- The bit that falls off one end is **reused** at the other end

Direction	What happens
Left	Leftmost bit moves to the rightmost position
Right	Rightmost bit moves to the leftmost position

- The following number is shifted by **three** places to the left and right

Left and Right

Step	128	64	32	16	8	4	2	1
Original	1	0	1	1	0	0	0	1
Cyclic left (×3)	1	0	0	0	1	1	0	1
Cyclic right (×3)	0	0	1	1	0	**1**	1	0

- The 3 leftmost bits 101 wrap around to the right side
- The 3 rightmost bits 001 wrap around to the left side
- **Original**: 1011 0001 = **177**
- **Cyclic left (3)**: 1000 1101 = **141**
- **Cyclic right (3)**: 0011 0110 = **54**

Shift type	Left shift	Right shift	Used for
Logical Shift	Shift bits left, fill with 0	Shift bits right, fill with 0	Unsigned numbers, raw bits

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Arithmetic Shift	Same as logical left	Shift bits right, **preserve sign bit**	Signed numbers (two's complement)
Cyclic Shift	Rotate bits left (no loss)	Rotate bits right (no loss)	Bit rotation, encryption, checksums



Your notes

- Cyclic shifts **do not** preserve the sign bit and are not equivalent to multiplying or dividing by powers of two

Device control & bit masking

What is device control?

- In computer systems and embedded devices, **each bit** in a byte can be used to represent the **state of a device or feature** (e.g. turning a light on, checking if a sensor is active, or flagging an error)
- You can **control or monitor** these bits using **bitwise operations** and **bit masking**

What is bit masking?

- Bit masking uses binary patterns (masks) to **test**, **set**, **clear**, or **toggle** specific bits **without affecting the others**

Bitwise AND operation (&)

- Used to **test** if a specific bit is set to 1
- Only returns 1 when **both** the binary value **and** the mask have 1 in the same position
- Useful for **checking** the status of a device

Description	**128**	**64**	**32**	**16**	**8**	**4**	**2**	**1**
Binary	1	0	1	1	1	0	0	1
Mask	0	0	1	1	0	0	0	0
Result	0	0	1	1	0	0	0	0

- Check if a **heater (Bit 5)** and **light (Bit 4)** are both ON
- Apply a mask with 1s at those positions
- If the result is **not 0**, the devices are **ON**

Bitwise OR operation (|)

- Used to **set** specific bits to 1
- Returns 1 if **either** the binary value **or** the mask has 1

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- This is useful to **turn on a device** (set a bit to 1) without changing the others

Description	128	64	32	16	8	4	2	1
Binary	1	1	0	0	1	0	1	0
Mask	0	1	1	1	0	0	0	0
Result	1	1	1	1	1	0	1	0

- Turn ON the **alarm system (Bit 6)** and **door lock (Bit 5)** by ORing with a mask that has 1s at those positions

Bitwise XOR operation (^)

- Used to **toggle** (flip) specific bits
- Returns 1 if the bits are **different**
- Can be used for **switching** a device from ON to OFF or vice versa

Description	128	64	32	16	8	4	2	1
Binary	1	0	1	0	1	0	1	0
Mask	0	0	1	1	0	0	0	0
Result	1	0	0	1	1	0	1	0

- Toggle the **fan (Bit 5)** and **heating system (Bit 4)** with a mask that has 1s in those positions

Summary

Operation	Symbol	Purpose	Example Use
Bitwise AND	&	Test a bit	Check if a sensor is active
Bitwise OR	|	Set a bit	Turn ON a device
Bitwise XOR	^	Toggle a bit	Flip device state (ON ↔ OFF)



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Operating Systems

Contents

- Operating System fundamentals
- Utility software
- Program libraries

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Purpose of an OS

What is an operating system (OS)?

- An operating system (**OS**) is software that **provides an interface between the user and the hardware** in a computer system
- An operating system **hides the complexities of the hardware** from the user, for example:
  - A user **does not need to know** 'where' on secondary storage data is kept, just that it is saved for when they want it again
- To help achieve this an OS provides a **user interface**

User interface

- A user interface is **how the user interacts with the operating system**
- Examples of user interfaces include:
  - **Command Line Interface (CLI)**
  - **Graphical User Interface (GUI)**
  - **Menu**
  - **Natural language (NLI)**



Copyright © Save My Exams. All Rights Reserved

What is a command line interface (CLI)?

- A Command Line Interface (**CLI**) requires users to interact with the operating system using **text-based commands**
- CLIs are more commonly used by **advanced users**
- Examples of CLIs are MS-DOS (Microsoft Disk Operating System) and Raspbian (for Raspberry Pi)

Study copy distributed by EduResource ZA | eduresourcza.com

What is a graphical user interface (GUI)?

- A Graphical User Interface (**GUI**) requires users to interact with the operating system using **visual elements** such as windows, icons, menus & pointers (**WIMP**)
- GUIs are **optimised for mouse and touch gesture input**
- Examples of GUIs are Windows, Android and macOS

What is a menu interface?

- A menu interface is **successive menus** presented to a user with a **single option at each stage**
- Often performed with **buttons** or a **keypad**
- Examples include:
  - **Chip and pin machines**
  - **Vending machines**
  - **Entertainment streaming services**

What is a natural language interface (NLI)?

- A natural language interface (**NLI**) uses the spoken word to respond to spoken or textual inputs from a user
- Examples include:
  - **Virtual assistants** - Amazon Alexa, Google Assistant, Siri
  - **Search engines**
  - **Smart home devices**

OS management tasks

An operating system has five key **management tasks**:

- **Memory management**
- **File management**
- **Security management**
- **Hardware management**
- **Process management**

Memory management

- Memory management is a process carried out by the operating system **allocating main memory (RAM) between different programs** that are **open at the same time**
- The OS is responsible for **copying programs and data from secondary to primary storage** as it is needed



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- **Programs and data require different amounts of RAM** to operate efficiently and the OS manages this process
- **RAM is allocated** based on **priority and fairness**, for example, system applications (essential) may have a higher priority than user applications
- The OS **dynamically manages the memory**, adjusting allocation as needed to maintain optimal system performance
- Memory management makes **multitasking** possible
  - Prevents data from different programs overwriting each other in RAM
  - Decides which processes and data should be kept in main memory at any given time



Your notes

File management

- File management is a process carried out by the operating system **creating, organising, manipulating and accessing files and folders on a computer system**
- The OS **manages where data is stored** in both **main memory** and **secondary storage**
- File management gives the user the ability to:
  - **Create files/folders**
  - **Name files/folders**
  - **Rename files/folders**
  - **Copy files/folders**
  - **Move files/folders**
  - **Delete files/folders**
- The OS allows users to control who can access, modify and delete files/folders (**permissions**)
- The OS provides a **search facility** to find specific files based on various criteria

Security management

- Operating systems provide various security features such as **password-protected system accounts**, a **firewall**, virus scanning and **file encryption**
- Password-protected system accounts are a very common feature in operating systems
- A system administrator is able to **allocate** different **access rights** for different users on a network
- The OS is able to **maintain settings** for individual users, such as desktop backgrounds, icons and colour schemes
- The OS **audits** (keeps a log of) files created by users, accesses, edits and deletes

Hardware management

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- Peripheral management is a process carried out by the operating system **managing the way peripherals (hardware) interact with software**
- The OS **allocates system resources** to peripherals to ensure **efficient operation**
- Peripheral management makes **plug-and-play** (**PnP**) functionality possible, **automatically detecting and configuring new peripherals** without the need for manually installing device drivers or power cycling the system
- A device driver is **a piece of software used to control a piece of hardware**
- Input/output devices **require device drivers** in order to be used by the operating system
- The OS has **generic device drivers built in** which makes **basic compatibility** possible
- In order for hardware to be used to its **maximum capacity**, often **a separate device driver must be downloaded** from the manufacturer
- Device drivers are **OS specific** and are **regularly updated**



Your notes

Process management

- Process management is a process carried out by the operating system **dividing time** (**time slicing**) into small chunks and **allocating them to different processes**
- The CPU executes **one instruction at a time**, but rapid context switching allows **multiple processes** to share processor time
- The OS uses a **scheduling algorithm** to **prioritise** processes
- Processes are **placed in queue** whilst waiting to be carried out, they return to the back of the queue when their time is up
- The goal of process management is to **share resources** (**CPU & main memory**)
- Allows processes to be **paused and resumed** using context switching
- Decides **which process is executed next** using a scheduling algorithm



Worked Example

Identify four key management tasks that the Operating System will perform. **[4]**

Answer

- Memory management [1 mark]
- File management [1 mark]
- Security management [1 mark]
- Hardware / device / peripheral / resources management [1 mark]
- Input/output management [1 mark]
- Process management [1 mark]
- Error checking and recovery [1 mark]
- Provision of a platform for software [1 mark]

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- Provision of a user interface [1 mark]



Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Utility software

What is utility software?

- Utility software is a **collection of tools** designed to help **maintain** a computer system
- Utility software is designed to **perform a limited number of tasks**
- Utility software **interacts with the computers hardware**, for example, secondary storage devices
- Some utility software comes **installed with the operating system**
- Examples of utility software include:
  - **Disk formatter**
  - **Virus checker**
  - **Defragmentation software**
  - **Disk contents analysis/disk repair software**
  - **File compression**
  - **Back-up software**

Disk formatter

- A disk formatter **prepares a storage device** (like a hard drive or USB stick) for use by **creating a file system** and **organising the space into sectors and tracks**
- It is needed:
  - To **wipe** and **re-initialise** a disk before use
  - To change the **file system format** (e.g. FAT32, NTFS)
  - To **remove all data** and errors before installing a new OS or reusing the drive
- For example:
  - **Formatting** a new external drive before saving files
  - **Reformatting** a corrupted USB stick to make it usable again

Virus checker

- Virus checking software is a **combination of different software to prevent** computers from being susceptible to **viruses** and other **malicious software**
- It typically includes:
  - **Anti-virus**

© 2026 Save My Exams, Ltd.

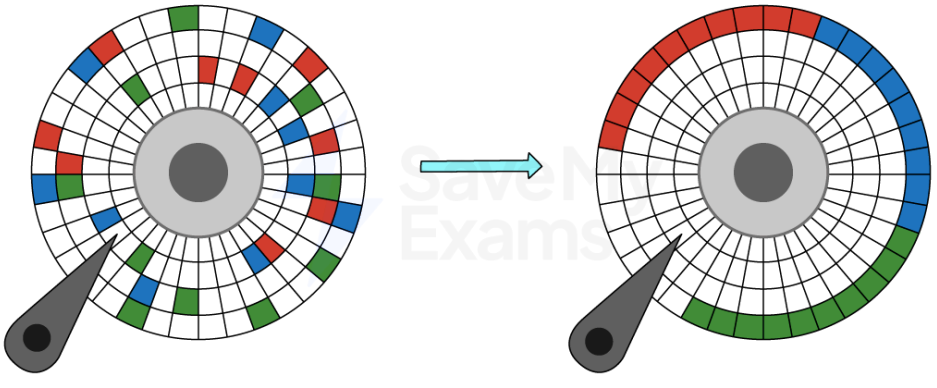
Study copy distributed by EduResource ZA | eduresourcza.com

- **Anti-spam**
- **Anti-spyware**

- A virus checker **scans** through **email** attachments, **websites** and downloaded **files** to search for issues
- Two ways virus checkers can approach the task are:
  - Use a list of known **unique malware fingerprints** (**signatures**) to block immediately if they try to access your device in any way
  - **Monitor** the **behaviour** of programs to identify suspicious activities that might indicate malware such as;
    - **Rapid deletion/modification of files**
    - **Attempts to access sensitive data/resources**
    - **Communicating with known malicious servers**
- Virus checkers will also perform **checks for updates** to ensure the database of signatures is **up to date**

Defragmentation software

- Defragmentation software **groups fragmented files back together** in order to **improve the access speed** of a hard disk drive **(HDD)**
- As programs and data are added to a new hard disk drive, it is added in order, **over time as files are deleted this leaves gaps**
- As programs and data are added over time, **these gaps get filled and data becomes fragmented**
- Defragmentation can only be used on **magnetic storage**



Copyright © Save My Exams. All Rights Reserved

Disk contents analysis/disk repair software

- This utility checks the **structure and health** of a disk, including **files, folders, sectors**, and the **file allocation table**

Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

- It is needed:
  - To **diagnose and fix errors** on the disk (e.g. bad sectors, file system corruption)
  - To **recover lost or damaged files**
  - To improve system performance by identifying and repairing issues
- For example:
  - Running **Check Disk (chkdsk)** on Windows to repair file system errors
  - Using a **disk utility** on macOS to verify and repair drive problems



Your notes

File compression

- Compression **reduces the amount of secondary storage** required by performing an algorithm on the original data
- **Lossy compression** physically **removes data** from the original data to reduce its size, **the original file can not be re-created**
- **Lossless compression** uses mathematics to **order data** more efficiently reducing its size, **the original files can be re-created as no data is lost**

Back-up software

- Back-up software is used to **create copies of personal data** in order to keep it safe in the event of:
  - **Accidental loss**
  - **Data theft**
- Backups can be **automated** and **scheduled** to happen at less busy periods of the day, to not take up valuable system resources (e.g. overnight etc.)
- Backups can be made in two ways:
  - **Full** - all files are backed up (saftest, slow)
  - **Incremental** - only files that have been added/modified since the last backup are backed up (faster, less secure)
- Backups can be stored **locally** (secondary storage) or **remotely** (cloud)
- Backup software can be **purchased** or come as a standalone application **bundled** with an **operating system**



Worked Example

Draw one line from each utility software to its correct description.

Utility software Description

© 2026 Save My Exams, Ltd.

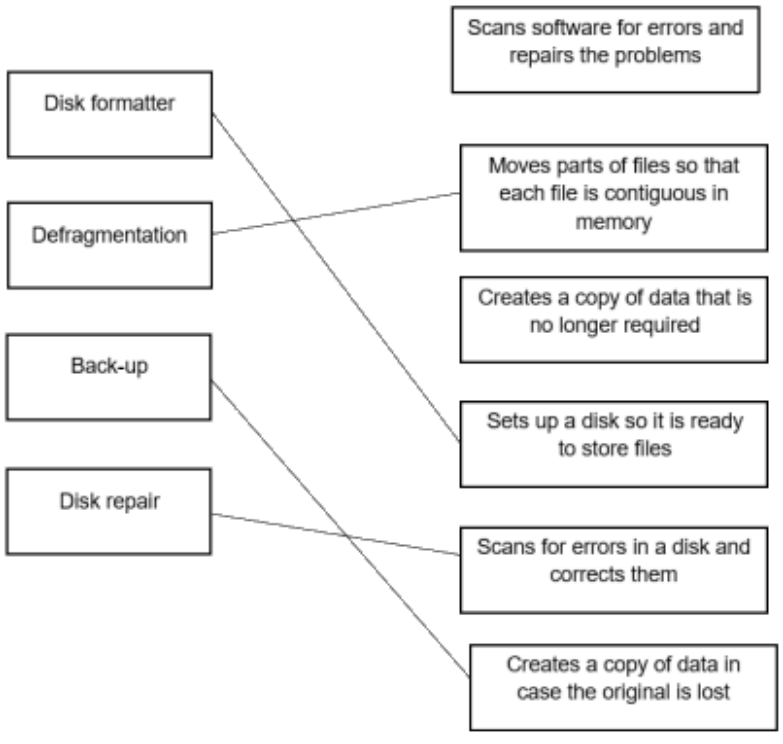
Study copy distributed by EduResource ZA | eduresourcza.com

	Scans software for errors and repairs the problems
Disk formatter	Moves parts of files so that each file is contiguous in memory
Defragmentation	Creates a copy of data that is no longer required
Back-up	Sets up a disk so it is ready to store files
Disk repair	Scans for errors in a disk and corrects them
	Creates a copy of data in case the original is lost

[4]

Your notes

Answer



Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Program libraries

What is a program library?

- A program library is a **collection of pre-written code** (called **library routines**) that programmers can **reuse** in their own software
- These routines can **perform common tasks** such as:
  - **Sorting**
  - **Displaying graphics**
  - **Playing sounds**
  - **Managing data**
- Advantages of using program libraries includes:

Advantage	Explanation
Saves development time	Pre-written routines can be reused, so programmers don't need to write common functions from scratch
Promotes code reuse	Well-tested routines can be used in multiple programs, reducing duplication
Improves reliability	Library routines are usually thoroughly tested and debugged, reducing the chance of errors
Easier maintenance	Updates to a library routine automatically benefit all programs that use it (especially with **DLLs**)
Efficient memory usage	**Dynamic link libraries (DLLs)** are only loaded when needed at runtime, saving system resources
Standardisation	Promotes consistency in how common tasks (e.g. file handling, sorting) are implemented

What is a DLL?

- A **Dynamic Link Library (DLL)** is a file that contains **pre-written code**, such as **functions or routines**, that can be **used by multiple programs** at **runtime**

Aspect	Advantages	Disadvantages

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Program Size	Reduces program size by keeping reusable code separate	-
Memory Usage	Saves memory by sharing DLLs between programs	-
Code Reuse	Encourages reuse of common routines (e.g. sorting, sound)	-
Modularity	Supports modular design for easier development and maintenance	-
Ease of Updates	Update a DLL once to improve all programs that use it	Can cause **version conflicts** if newer DLLs aren't compatible
Dependencies	External DLLs can be reused by many applications	Program may **fail to run** if a required DLL is **missing**
Security	-	Malicious or altered DLLs can pose **security risks**
Debugging	-	Errors in DLLs can be **hard to trace** across multiple programs



Worked Example

Jennifer is writing a computer program for her A Level homework.

Jennifer uses a program library to help her write her computer program.

Describe how a program library can be used while writing a computer program. **[2]**

Answer

- Program libraries store pre-written functions and routines [1 mark]
- The program library can be referenced/imported [1 mark]
- The functions/routines can be called in her own program [1 mark]



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Language Translators

Contents

- Translator types
- Development tools

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

What is a translator?

- A translator is a program that **translates** program **source code** into **machine code** so that it can executed directly by a processor
- **Low-level languages** such as **assembly code** are translated using an **assembler**
- **High-level languages** such as **Python** are translated using a **compiler** or **interpreter**

Assembler, compiler, interpreter

What is an assembler?

- An assembler translates **mnemonics written in assembly language** (low-level) in **to machine code**
- **Each lime** of assembly language is **assembled** into a **single machine code** instruction
- Assemblers have been used less and less since high-level languages were introduced

Advantages	Disadvantages
Speed of execution	Difficult to write due to limited and hard to understand commands
Optimises the code	Changes mean it must be reassembled
Original source code will not be seen	Designed solely for one specific processor

What is a compiler?

- A compiler translates high-level languages into machine code **all in one go**
- Compilers translate inputs into machine code **directly**
- Compilers are generally used when **a program is finished** and has been checked for syntax errors
- Compiled code can be **distributed (creates an executable)** and run **without the need for translation software**
- If compiled code contains any errors, after fixing, it **will need re-compiling**

Advantages	Disadvantages
Speed of execution	Can be memory intensive

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

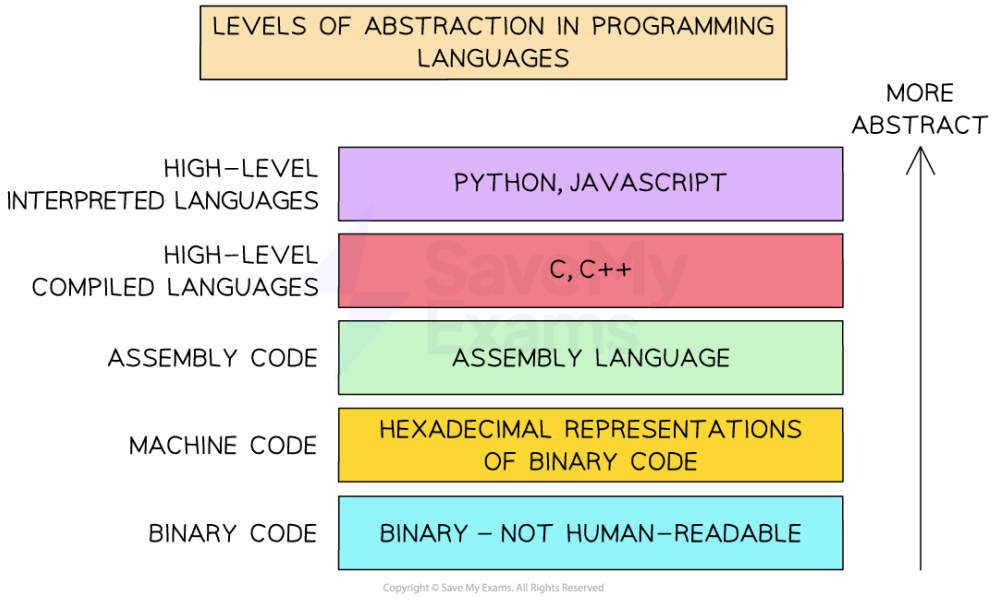
Optimises the code	Difficult to debug
Original source code will not be seen	Changes mean it must be recompiled
	It is designed solely for one specific processor

Your notes

What is an interpreter?

- An interpreter translates high-level languages into machine code **one line at a time**
- Each line is executed after translation and if **any errors are found**, the **process stops**
- Interpreters **do not** generate machine code **directly**, appropriate machine code subroutines are called
- Interpreters are generally used when a program **is being written** in the development stage
- Interpreted code is more **difficult to distribute** as **translation software is needed for it to run**

Advantages	**Disadvantages**
Stops when it finds a specific **syntax error** in the code	Slower execution
Easier to debug	Every time the program is run it has to be translated
Require less RAM to process the code	Executed as is, no optimisation



Study copy distributed by EduResource ZA | eduresourcza.com

Compiler vs interpreter

Aspect	Compiler	Interpreter
Software needed by user	End users only need the **executable file**, not the compiler – so there's **no extra cost or setup**	End users **need an interpreter** or compiler to translate and run the source code – may involve setup or cost
Access to source code	Users **do not receive the source code**, so they **can't modify or extend** the program	Users receive the **full source code**, making it **easier to modify or extend** the program themselves
Control and monetisation	Developers keep control of the code and can **charge for upgrades** or customisations	Developers lose control over their code, making it **harder to monetise upgrades** or prevent copying
Speed of execution	Compiled programs **run faster**, as translation is done in advance and optimised machine code is used	Interpreted programs **run slower**, since each line is translated every time it runs
Error handling	Compiled programs don't run until **all syntax and semantic errors** are fixed	Errors are **caught one line at a time**, making it easier for the developer to test and fix during development
Portability	Compiled code can be **run on different machines**, regardless of where it was compiled	Interpreted programs must be interpreted on the **same type of machine** – less portable
Debugging experience	More difficult to test sections – developers need to **write special routines** to check partial results	Developers can **view partial results as they go**, making it easier to test, debug, and refine ideas
Crash risk	Untested compiled programs may **crash the system** if run with errors	Interpreted programs are **safer during testing** – errors are caught before the system can crash
User independence	Users are **reliant on the developer** for updates or modifications – they can't edit the code themselves	Users have **full access to code and libraries**, giving them **freedom to customise or fix** the program



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Mixed mode translation

What is mixed mode translation?

Your notes

- Mixed mode translation **combines features** of both a **compiler** and an **interpreter**
- Instead of fully compiling or interpreting the code, the program is **partially compiled into an intermediate form**, which is then **interpreted at runtime**
- It works by:
  - The source code is **compiled into intermediate code**, not full machine code
  - This **intermediate code** (e.g. bytecode) is **interpreted by a virtual machine** (e.g. the Java Virtual Machine – JVM)
  - Optionally, some parts may later be **just-in-time (JIT) compiled** into machine code for better performance
- **Java** uses mixed mode translation:
  - Java source code is **compiled** to **bytecode** (.class file)
  - Bytecode is **interpreted** (or JIT compiled) by the **JVM** on any platform

Feature	Compiler	Interpreter	Mixed mode translation
Translation	Entire program at once	One line at a time	Compiles to intermediate code
Speed of execution	Fastest (fully compiled)	Slowest	Medium (JIT makes it faster over time)
Portability	Not portable	Not portable	Highly portable (across platforms)
Errors shown	After full compilation	One at a time	During compile or runtime
Requires runtime system	No	Yes	Yes (e.g. JVM)
Example languages	C, C++	Python, JavaScript	Java, C# (with .NET CLR)

Worked Example

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Jennifer uses an Integrated Development Environment (IDE) to write her computer program.

The IDE allows Jennifer to use both an interpreter and a compiler while creating her computer program.

Describe the ways in which Jennifer can use both a compiler and an interpreter while developing the program. **[4]**

Answer

Interpreter:

- Use an interpreter while writing the program [1 mark]
- … to test/debug the partially completed program [1 mark]
- … because errors can be corrected and processing continue from where the execution stopped // errors can be corrected in real time // errors are identified one at a time [1 mark]

Compiler:

- Use the compiler after the program is complete [1 mark]
- … to create an executable file [1 mark]
- Use the compiler to repeatedly test the same (completed) section [1 mark]
- … without having to re-interpret every time // compiler not needed at runtime [1 mark]



Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

What is an IDE?

- An **Integrated Development Environment (IDE)** is a software tool that provides programmers with a comprehensive and integrated platform to **write**, **edit**, **compile**, **debug**, and **manage** their code efficiently
- IDEs are designed to **streamline the software development process** by offering a wide range of features and tools that can help programmers
- IDEs can be an **app to download** or can be **web-based**

IDE features

Syntax highlighting (pretty printing)

- IDEs use syntax highlighting to **visually distinguish** different elements of code based on their syntax
- The IDE assigns **distinct colours or styles** to keywords, strings, comments, variables, functions, and other language constructs
- Syntax highlighting helps programmers **quickly identify and differentiate code** components, reducing the chance of syntax errors and improving code comprehension
- In this example Python code, you can see that the code comments are displayed in red, the strings are displayed in green and the print commands are displayed in purple

```
# global variable
global_var = "I am a global variable"

def modify_global_var():
    # declaring variable as global
    global global_var
    # modifying global variable
    global_var = "I have been modified"

print(global_var)  # outputs: I am a global variable
modify_global_var()
print(global_var)  # outputs: I have been modified
```

Syntax highlighting in an IDE

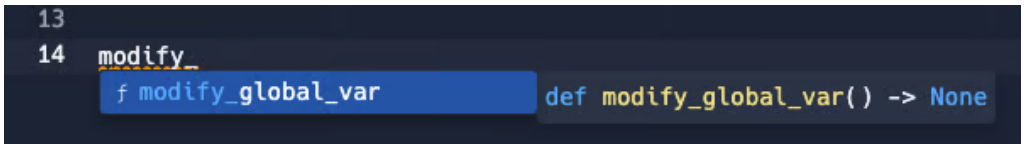
Autocomplete

- Autocomplete will **suggest code completions** as programmers type
- For example, as developers start typing a variable, function, or method name, the IDE displays a list of available options

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

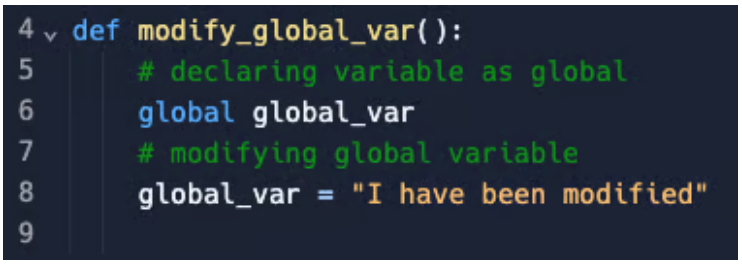
- Another example is when programmers use opening brackets and the autocomplete facility will automatically add closing brackets
- Code completion **improves productivity** by reducing typing effort and preventing typing errors and naming inconsistencies
- IDEs can automatically correct common mistakes or typing errors made during coding. When an error-prone sequence is detected, **the IDE can suggest a correction**, helping developers fix minor errors as they type their code



Code completion in an IDE

Auto indent

- Auto indent will **automatically indent the programming code** when you start a new line of code within a code block
- They are commonly used in **selection** and **iteration** programming constructs
- In the example below, the indentation is automatic after starting to define the code block on line 04 and is clearly marked to show which lines of code are included within the block.



Code formatting with automatic indentation

Code comments

- Comments are **lines of text within the code that are not executed** but serve as documentation and explanations for programmers
- IDEs enable developers to add single-line or multi-line **comments to describe the code's purpose**, algorithmic steps, and any relevant information that will be useful to the maintenance of the program
- Comments enhance code understanding, making it easier for developers to revisit and modify code at a later stage
- Comments are in green on the image above

Syntax error highlighting

Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

- IDEs can automatically highlight **syntax errors in the code editor** as developers type
- This feature **helps catch mistakes instantly**, allowing programmers to address syntax-related issues promptly
- Syntax errors are highlighted with distinct colours, making them easily noticeable, even before executing the code



Your notes

Variable watch window

- The watch window is a debugging tool that allows developers to **monitor the values of specific variables** during **runtime**
- Developers can add variables to the watch window and observe their values change as the program executes
- This tool is particularly valuable for understanding how variables evolve throughout the program's execution and diagnosing issues related to variable values

Breakpoints

- Breakpoints are **markers** that developers can set at specific lines of code to **halt program execution** during **runtime**
- When the program reaches a breakpoint, it pauses, allowing developers to inspect the program's state and variable values and identify the cause of potential bugs
- Breakpoints provide a controlled way to analyse code step-by-step

Error message list

- The error message list **displays a collection of errors**, **warnings**, and **messages** generated during the compilation and execution of the program
- IDEs present detailed error descriptions and relevant code locations, enabling programmers to address and correct issues systematically

Stepping mode

- Stepping mode is a debugging mode that allows developers to **execute the program one line at a time**
- Developers can choose between different step options, such as step into, step over, and step out, to examine function calls and control flow
- Step mode offers a granular view of the program's execution, making it easier to identify the exact location of potential bugs

Crash dump/post-mortem report

- When a program crashes, IDEs can **generate** a crash dump or post-mortem report
- This report **captures the program's state** and **memory** contents **at the time of the crash**, helping developers diagnose the root cause of the crash after the program has terminated unexpectedly

Stack contents

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- The stack contents tool allows developers to **inspect the contents of the program's call stack** during debugging
- It shows the order in which functions were called and the parameters and local variables of each function
- Understanding the call stack, aids in understanding the program's flow and diagnosing issues related to function calls



Your notes



Worked Example

Identify two debugging tools that a typical IDE can provide. **[2]**

Answer

e.g.

- Breakpoints [1 mark]
- Single stepping [1 mark]
- Report windows [1 mark]

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Data Security

Contents

- Security measures
- Threats
- Data protection

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Security, privacy, integrity

What is the difference between data security, privacy and integrity?

Term	Definition	Key focus	Example
Data Security	Protecting data from **unauthorised access**, theft, or attacks	Preventing breaches or leaks	Using encryption, firewalls, and passwords to protect stored data
Data Privacy	Ensuring data is **collected, stored, and shared** in a way that respects the user's rights	Controlling who can access and use personal data	Asking for user consent before collecting or sharing personal information
Data Integrity	Ensuring data is **accurate, complete, and unaltered** during storage or transfer	Maintaining correctness and reliability	Using checksums or validation rules to detect accidental or unauthorised changes

- **Security** = keeping data safe from threats (e.g. hackers)
- **Privacy** = making sure data is used **fairly** and **with consent**
- **Integrity** = making sure data stays **accurate** and **unchanged**

Protection techniques

User accounts

- User accounts are designed to **control access** to computer systems
- They help protect data and system resources by ensuring only **authorised individuals** can log in and perform specific tasks

Passwords

- Passwords are a **digital lock** to prevent unauthorised access to an account
- They are often stored as an **encrypted/ciphered** text entry in a database, ensuring that even with unauthorised access to a database, a hacker would not be able to gain access to the individual passwords of users

Authentication

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- Authentication is the process of **ensuring that a system is secure** by asking the user to **complete tasks to prove they are an authorised** user of the system
- Authentication is done because **bots can submit data in online forms**
- Authentication can be done in several ways, including:
  - **Digital signatures**
  - **Biometrics**



Your notes

Digital signatures

- A digital signature is a secure way to **prove that a digital message or document was sent by a specific person** and that it **has not been altered**
- It acts like a **digital stamp of approval**, confirming:
  - **Who sent the data** (authentication)
  - **That the data hasn't been changed** (integrity)

Biometrics

- Biometrics are an individuals **personal characteristics used to identify them**, such as
  - **Fingerprints**
  - **Iris/retina scans (eyes)**
  - **Voice recognition**
- Biometrics provide a **very secure method** of confirming a users **identity** before allowing access/permission to a computer system
- Biometric measures are often **used on mobile devices** to provide secure access

Firewall

- A firewall is a **barrier** between a network and the internet
- A firewall prevents **unwanted traffic** from entering a network by filtering requests to ensure they are **legitimate**
- It can be both **hardware** and **software** and they are often used together to provide stronger security to a network
  - **Hardware firewalls** will protect the whole network and prevent unauthorised traffic
  - **Software firewalls** will protect the individual devices on the network, monitoring the data going to and from each computer

Anti-virus software

- Anti-virus software is a term used to describe a combination of different software to prevent computers from being susceptible to **viruses** and other **malicious software**

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- Anti-virus **scans** through **email** attachments, **websites** and downloaded **files** to search for issues
- Anti-virus software has a list of known malware **signatures to block** immediately if they try to access your device in any way
- Anti-virus will also perform **checks for updates** to ensure the database of known issues is up to date



Your notes

Anti-spyware software

- Anti-spyware is a type of **security software** designed to **detect, block, and remove spyware** from a computer system
- Spyware is a type of **malicious software (malware)** that secretly gathers information about a user without their knowledge
- It can:
  - **Record keystrokes** (e.g. passwords, credit card numbers)
  - **Monitor browsing habits**
  - **Access files** and send them to a third party

Encryption

- Encryption is the process of **converting data into a secret code** so that **only authorised users** can read it
- It protects sensitive information (like passwords, personal data, or messages) from **unauthorised access**, especially when data is stored or sent over a network

Type	Description
Symmetric	The **same key** is used to encrypt and decrypt. Fast but key must be shared securely
Asymmetric	Uses a **public key** for encryption and a **private key** for decryption. More secure for sending data



Worked Example

A company has several security measures in place to prevent unauthorised access to the data on its computers.

Describe the difference between the security and privacy of data. **[2]**

Answer

- Security protects data against loss [1 mark]

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- Privacy protects data against unauthorised access [1 mark]



Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Network & internet risks

What are common network & internet risks?

Hackers

- A **hacker** is someone who gains **unauthorised access** to computer systems or networks
- Hackers often **exploit security weaknesses** to:
  - **Steal sensitive data**
  - **Gain control of systems**
  - **Cause damage or disruption**
- Hackers are considered **cybercriminals** when their actions are illegal or harmful.

How do hackers gain access?

- Hackers look for opportunities or vulnerabilities in systems, such as:
  - **Unpatched software** – Missing security updates can leave systems exposed
  - **Out-of-date anti-malware** – Older protection can fail to detect new threats
  - **Weak passwords** – Simple or reused passwords are easy to guess or crack

Effects of a hacker attack

- When a hacker successfully gains access, it can lead to:

Effect	Impact
Data breaches	Personal or company information is leaked or stolen
Malware installation	Hackers may install viruses or spyware to cause further harm
Data loss	Important files may be deleted or corrupted
Identity theft	Stolen personal data can be used to impersonate individuals
Financial loss	Hackers may access bank accounts or demand ransom payments

How can hacking be prevented?

- There are several methods to reduce the risk of being targeted by hackers:
  - Use **strong passwords** that are long and hard to guess

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- Enable **two-factor authentication** for extra login security
- Install and regularly update **anti-malware software**
- Use **firewalls** to block unauthorised access to the network
- Keep all **software up to date** with security patches



Your notes

Phishing

- Phishing is a form of social engineering
- It involves sending **fraudulent, legitimate-looking emails** to a large number of email addresses, claiming to be from a **reputable company** or trusted source to try and **gain access** to your details
- Phishing often tries to **coax** the user to click on a login button to enter their details

What are the effects of phishing?

- The creator of the email can gain unauthorised access to **personal data** such as **login information**, **bank accounts** and more
- Phishing can lead to **identity theft** or **fraudulent activity** on credit cards and bank accounts

How can phishing be prevented?

- Phishing can be prevented by:
  - **Anti-spam filters** to avoid fraudulent emails arriving in a user's inbox
  - **Training staff** to recognise fraudulent emails and to avoid opening attachments from unrecognised senders
  - **User access levels** to prevent staff from being able to open files-types such as executable (**.exe)** files and batch (**.bat**) files

Pharming

- Pharming is typing a website address into a browser and it is **redirected to a 'fake' website** to trick a user into typing in sensitive information such as passwords
- An attacker attempts to **alter DNS** settings or **change a users browser settings** to redirect users to the fraudulent website

What are the effects of pharming?

- The creator of the malicious content can gain unauthorised access to **personal data** such as **login information, bank accounts** and more
- Pharming can lead to **identity theft** or **fraudulent activity** on credit cards and bank accounts

How can pharming be prevented?

- Pharming can be prevented by:

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- **Keeping anti-malware software up to date**
- **Checking URLs regularly**
- **Make sure the padlock icon is visible**



Your notes

Malware

- Malware (**mal**icious soft**ware**) is the term used for any software that has been created with malicious intent to cause harm to a computer system
- Examples of issues caused by malware include
  - Files being **deleted**, **corrupted** or **encrypted**
  - Internet connection becoming **slow** or **unusable**
  - Computer **crashing** or **shutting down**
- Malware can exist in many forms, each designed to perform its role in different ways

Malware	What it Does
Computer virus	▪ A program which can **replicate itself** on a user's computer. It contains code that will cause **unwanted and unexpected events** to occur ▪ Examples of issues a user may experience are ▪ **Corrupt** files ▪ **Delete** data ▪ **Prevent** applications from running correctly
Trojan	▪ Sometimes also called a **Trojan Horse** ▪ Trojans **disguise** themselves as **legitimate software** but contain malicious code in the background
Spyware	▪ Software which will allow a person to **spy** on the users' **activities** on their devices ▪ This form of software will be embedded into other software such as games or programs that have been downloaded from **illegitimate sources** ▪ Spyware can **record** your screen, log your **keystrokes** to gain access to **passwords** and more

How can malware be prevented?

- To protect against the threat of malware:
  - **Ensure code is written correctly**
  - **Keep anti-malware software up to date**

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- Install a firewall
- Educate users



Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Data security techniques

Access rights

- Access rights control **what users can see or do** on a network
- They ensure that users can **access the files and resources they need** for their role, and **can't access information they shouldn't**
- Access rights are assigned based on a user's **role**, **responsibilities**, or **security clearance**

Access right	What it means
Full access	The user can **open, create, edit, and delete** files or folders
Read-only access	The user can **open and view** files but **cannot edit or delete** them
No access	The user **cannot see or interact with** the file or folder at all

Example: access rights on a school network

User group	Access rights
Administrators	**Unrestricted access** – Can access and control all files, folders, and settings
Teaching Staff	**Partially restricted access** – Can access student data but not other staff files
Students	**Restricted access** – Can only access **their own files and folders**

- Users can be **grouped** (e.g. "Year 11", "Staff", "Admin") and given access rights as a group
- Access rights can be assigned to **files**, **folders**, or **whole systems**
- This helps protect **confidential data**, improves **network security**, and supports **efficient collaboration**

Encryption

- Encryption is a method of **scrambling data** before being transmitted across a network in order to protect the contents from **unauthorised access**

© 2026 Save My Exams, Ltd.

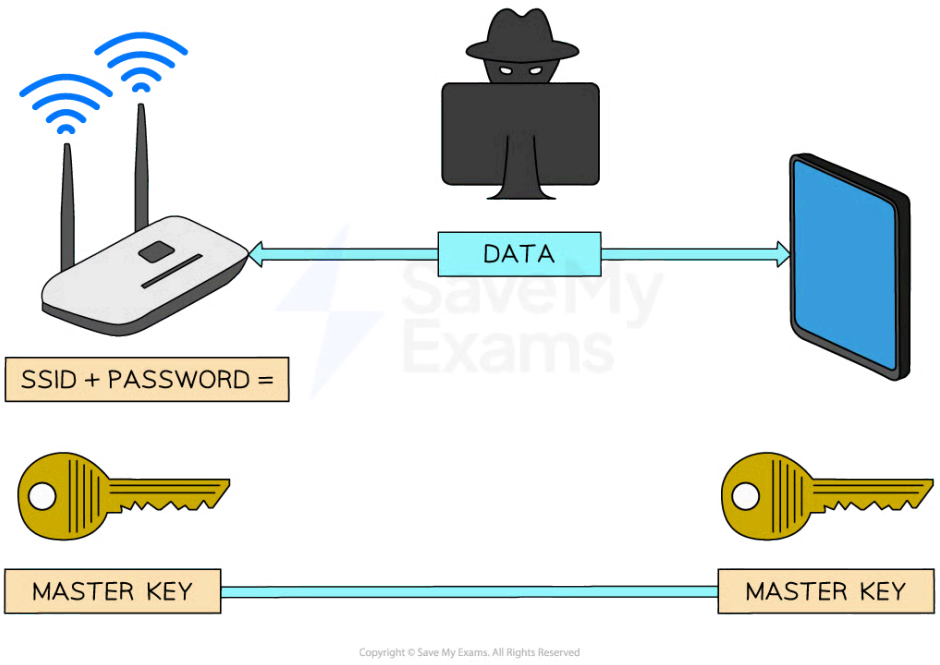
Study copy distributed by EduResource ZA | eduresourcza.com

- While encryption is important on both **wired** and **wireless** networks, it's even more critical on wireless networks due to the data being transmitted over radio waves, making it easy to **intercept**

Your notes

How is wireless data encrypted?

- Wireless networks are identified by a 'Service Set Identifier' (**SSID**) which along with a password is used to create a '**master key**'
- When devices connect to the **same wireless network** using the SSID and password they are given a copy of the master key
- The master key is used to **encrypt** data into '**cipher text**', before being transmitted
- The receiver uses the same master key to **decrypt** the cipher text back to '**plain text**'
- To guarantee the security of data, the **master key is never transmitted**. Without it, any intercepted data is rendered useless
- Wireless networks use dedicated **protocols** like **WPA2** specifically designed for **Wi-Fi** security



Copyright © Save My Exams. All Rights Reserved

How is wired data encrypted?

- Wired networks are encrypted in a very similar way to a wireless network, using a **master key to encrypt** data and the **same key to decrypt** data
- Encryption on a wired network differs slightly as it is often left to **individual applications to decide** how encryption is used, for example **HTTPS**

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Data Integrity

Contents

- Integrity methods

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Validation & verification

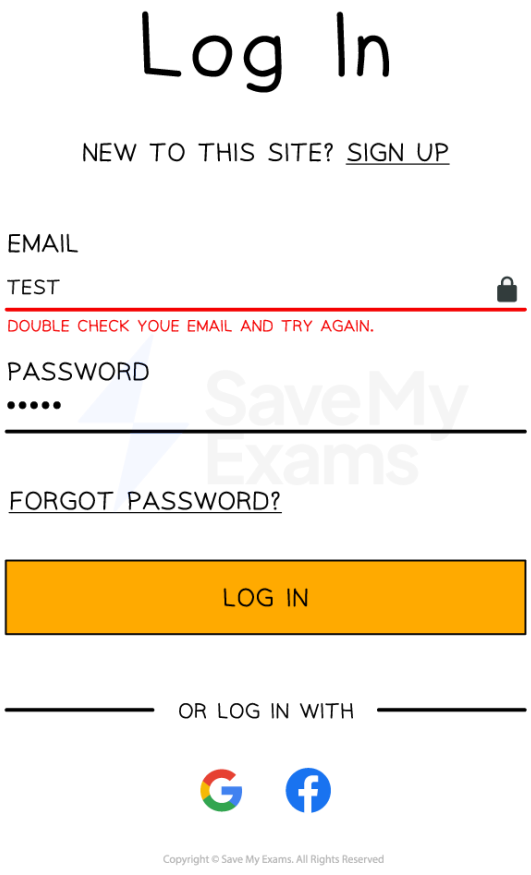
What is validation?

- Validation is an **automated process** where **a computer checks** if a user input is **sensible** and meets the program's requirements
- There are **seven categories** of validation which can be carried out on **fields** and **data types,** these are
  - **Range check**
  - **Format check**
  - **Length check**
  - **Presence check**
  - **Existence check**
  - **Limit check**
  - **Check digit**
- There can be occasions where **more than one type of validation will be used** on a field
- An example of this could be a password field which could have a **length**, **presence** and **type** check on it

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes



What is verification?

- Verification is the **act of checking data is accurate** when being **transferred** or **entered** into a system
- Verification methods include:
  - **Parity check (transfer)**
  - **Checksum (transfer)**
  - **Double entry checking (entry)**
  - **Visual checks (entry)**

Validation methods

Validation type	Definition	Example
Range check	Ensures a number falls within a set range	Validating that a percentage is between 0 and 100

Study copy distributed by EduResource ZA | eduresourcza.com

Limit check	Checks that a value does not exceed a maximum limit	Ensuring no more than 5 items can be bought in a special offer
Length check	Checks the length of a string	Confirming a PIN is exactly 4 digits long
Type check	Checks that the input is of the correct data type	Ensuring age is entered as a whole number
Presence check	Checks that data has been entered and the field is not blank	Making sure a registration form is not submitted with blank fields
Existence check	Checks that a referenced value exists in a database or list	Confirming a student ID entered exists in the school records
Format check	Ensures data is in the correct format (e.g. patterns)	Making sure an email address contains '@' and a domain like '.com'



Your notes

What is a check digit?

- A check digit is the **last digit** included in a code or sequence, used to **detect errors** in **numeric data entry**
- Examples of errors that a check digit can help to identify are:
  - **Incorrect digits entered**
  - **Omitted or extra digits**
  - **Phonetic errors**
- Added to the **end of a numerical sequence** they ensure **validity** of the data
- Calculated using **standardised algorithms** to ensure widespread **compatibility**
- Examples of where check digits can be used include:
  - **ISBN book numbers**
  - **Barcodes**

ISBN book numbers

- Each book has a **unique** ISBN number that **identifies** the book
- A standard ISBN number may be ten digits, for example, 965–448–765–9
- The check digit value is the **final digit** (9 in this example).
- This number is chosen specifically so that when the algorithm is completed the result is a whole number (an integer) with no remainder parts

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- A check digit algorithm is performed on the ISBN number and **if the result is a whole number,** then the **ISBN is valid**



Your notes

Barcodes

- Barcodes consist of **black and white lines** which can be scanned using barcode scanners
- Barcode scanners **shine a laser** on the black and white lines which **reflect light** into the scanner
- The scanner reads the **distance between these lines as numbers** and can identify the item
- Barcodes also use a **set of digits** to **uniquely identify each item**
- The **final digit** on a barcode is usually the **check digit**, this can be used to **validate** and **authenticate** an item

Verification methods

What is a parity check?

- A parity check determines whether **bits in a transmission** have been **corrupted**
- Every byte transmitted has **one of its bits** allocated as a **parity bit**
- The sender and receiver must **agree** before transmission whether they are using **odd** or **even parity**
- If **odd parity** is used then there must be an **odd number of 1's in the byte**, including the parity bit
- If **even parity** is used then there must be an **even number of 1's in the byte**, including the parity bit
- The **value of the parity bit** is determined by **counting the number of 1's in the byte**, including the parity bit
- If the number of 1's **does not match the agreed parity** then an **error has occurred**
- Parity checks **only check that an error has occurred**, they do not reveal where the error(s) occurred

Even parity

- Below is an arbitrary binary string

EVEN Parity bit	Byte						
0	1	0	1	1	0	1	0

- If an **even parity bit** is used then **all bits** in the byte, including the parity bit, must **add up to an even number**

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- There are four 1's in the byte
- This means the parity bit must be 0 otherwise the whole byte, including the parity bit, would add up to five which is an odd number



Your notes

Odd parity

- Below is an arbitrary binary string

ODD Parity bit	Byte						
1	1	0	1	1	0	1	0

- If an **odd parity bit** is used then **all bits** in the byte, including the parity bit, must **add up to an odd number**
  - There are four 1's in the byte. This means the parity bit must be a 1 otherwise the whole byte, including the parity bit, would add up to four which is an even number
- The table below shows a number of examples of the agreed parity between a sender and receiver and the parity bit used for each byte

Example #	Agreed parity	Parity bit	Main bit string							Total number of 1's
#1	ODD	0	1	1	0	1	0	1	1	5
#2	EVEN	1	0	0	0	1	0	0	0	2
#3	EVEN	1	0	1	0	1	1	1	1	6
#4	ODD	1	0	1	1	1	0	0	1	5
#5	ODD	1	1	0	1	0	1	0	1	5
#6	EVEN	0	1	0	0	1	1	1	0	4

- Example #1: The agreed parity is **odd**. All of the 1's in the main bit string are added **(5)**. As this number is **odd already** the **parity bit** is set to **0** so the whole byte **stays odd**
- Example #2: The agreed parity is **even**. All of the 1's in the main bit string are added **(1)**. As this number is **odd** the **parity bit** is set to **1** to make the **total number of 1's even (2)**
- Example #6: The agreed parity is **even**. All of the 1's in the main bit string are added **(4)**. As this number is **even already** the **parity bit** is set to **0** so the whole byte **stays even**

How do errors occur?

- When using parity bits, an **error** occurs when the number of **total bits does not match the agreed parity**

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- Bits can be **flipped** or **changed** due to **interference** on a wire or wirelessly due to **weather** or **other signals**

Example #	Agreed parity	Parity bit	Main bit string							Total number of 1's	Error
#1	ODD	1	1	1	0	1	0	1	1	6	Error
#2	EVEN	1	0	0	0	1	0	0	0	2	No error
#3	EVEN	1	0	1	1	1	1	1	1	7	Error
#4	ODD	1	0	1	1	1	0	0	1	5	No error
#5	ODD	1	1	0	1	0	1	1	1	6	Error
#6	EVEN	0	1	0	0	0	1	1	0	3	Error

- Parity checks are **quick** and **easy to implement** but **fail to detect bit swaps** that cause the parity to remain the same

What are parity byte & block checks?

- Parity blocks and parity bytes can be used to **check an error has occurred** and **where the error is located**
- Parity checks on their own **do not pinpoint where errors in data exist**, only that an error has occurred
- A parity block consists of **a block of data** with the number of 1's totalled **horizontally** and **vertically**
- A parity byte is also sent with the data which contains the **parity bits** from the **vertical parity calculation**
- Below is a parity block with a parity byte at the bottom and a parity bit column in the second column

ODD	Parity bit	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8
Byte 1	0	1	1	0	1	0	1	1
Byte 2	0	0	0	0	1	0	0	0
Byte 3	1	0	1	0	1	1	1	1



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Byte 4	1	0	1	1	1	0	0	1
Byte 5	1	1	0	1	0	1	0	1
Byte 6	1	1	0	0	1	1	1	0
Byte 7	0	0	1	1	1	1	1	0
Byte 8	0	1	0	1	1	0	0	0
Parity byte	0	1	1	1	1	1	1	1



Your notes

- The above table uses **odd parity**
- Each **byte row** calculates the **horizontal parity** as a parity bit **as normal**
- Each **bit column** calculates the **vertical parity** for each row, the **parity byte**
- It is **calculated before transmission** and **sent with the parity block**
- Each **parity bit tracks if a flip error** occurred in a byte while the **parity byte** calculates if an **error occurred in a bit column**
- By **cross referencing** both **horizontal and vertical** parity values the **error can be pinpointed**
- In the above example the **byte 3 / bit 5 cell** is the error and **should be a 0 instead**
- The error could be **fixed automatically or a retransmission request could be sent** to the sender

What is a checksum?

- A checksum is a value that can be used to determine if data has been **corrupted or altered**
- It indicates whether data **differs from its original form** but **does not** specify where
- Checksums are **calculated using an algorithm** and the value is **added to the transmission**
- The receiving device **re-calculates** the checksum and **compares** to the original
- If the **checksums do not match**, it is assumed an error has occurred

What is double entry checking?

- Double entry checking involves **entering the data twice** in separate input boxes and then comparing the data to ensure they both match
- If they do not, an error message is shown

What is a visual check?

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- A visual check involves the user **visually checking the data on the screen**
- A popup or message then **asks if the data is correct** before proceeding
- If it isn't the **user then enters the data again**



Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Ethics & Ownership

Contents

- Professional ethics
- Legal frameworks
- Artificial intelligence (AI)

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Ethics as a computing professional

What are ethics?

- Ethics are a set of principles that help us decide what is **right or wrong**
- They guide **how we behave and make decisions**, especially in situations where there isn't a clear answer
- In computing, an **ethical issue** is when technology creates a situation that challenges a person's **morals and values**
- These issues can be difficult because:
  - Technology moves fast, but **laws take time to catch up**
  - There may be **no single "correct" answer**
  - Choices can have **serious consequences** for both individuals and society

Computer ethics

- Computer ethics are a set of **principles that guide how computers and technology should be used responsibly and fairly**
- They help **prevent misuse** and **protect people's rights** in a digital world
- The three key areas of computer ethics are:
  - **Intellectual property rights**
    - It is **unethical to copy or use software, music, or digital content** without permission from the owner
    - Example: Downloading and using paid software without a licence
  - **Privacy**
    - Users have the right to keep their **personal information safe**
    - It is **unethical (and illegal)** to access someone's personal data without their permission (e.g. **hacking**)
  - **Impact on society**
    - Computers affect **how we work and live**
    - Ethical concerns include **job losses due to automation**, **social media effects**, and **digital addiction**
- The internet has made **plagiarism** easier, this is when someone **copies another person's work and claims it as their own**
- It's **not wrong to use someone's idea**, but you **must give credit**

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- This is done through:
  - **Footnotes**
  - **References or citations** at the end of a document
- **Anti-plagiarism software** exists to scan and detect copied content from websites



Your notes

Ethical bodies

What is an ethical body?

- An ethical body is an organisation that publishes a **code of conduct** which its members are expected to follow
- Being a member of an ethical body shows a commitment to **professional integrity** and maintaining **high ethical standards** in your work
- Examples of ethical bodies include:
  - **British Computer Society (BCS)**
  - **Institute of Electrical and Electronics Engineers (IEEE)**

British Computer Society (BCS)

- The **professional body for IT and computing** in the UK
- Promotes **ethical behaviour**, **good practice**, and **high standards** in tech-related jobs
- Members follow a **Code of Conduct** to ensure responsible use of technology

Institute of Electrical and Electronics Engineers (IEEE)

- A global organisation for **engineers, developers, and computer scientists**
- Sets **technical standards** and encourages **ethical and professional behaviour** in tech fields
- Known for publishing **research** and **codes of ethics**

Benefits of being a member of an ethical body

Benefit	Explanation
Clear ethical guidelines	Members follow a written code, so they don't have to decide what is ethical on their own
Trusted by clients and colleagues	Clients and staff know the member is working to recognised standards
Professional reputation	Being part of an ethical body shows you are reputable and serious about your work

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Recognised skills and knowledge	Entry may require qualifications or passing a test, showing you meet high standards
Help and support	Members can get advice (e.g. legal help) when needed
Training and development	Access to training courses helps members keep their skills up to date



Your notes

Ethical conduct

Example 1: Software development team

- A **project manager** leading a team of developers should:
  - **Treat all team members fairly**, listen to their ideas, and avoid favouritism
    - This helps team members feel **valued**.
  - **Respect work-life balance** by avoiding unrealistic deadlines
    - This supports better **teamwork** and helps get the **best work** out of them
  - Encourage **collaboration** and clear communication to avoid conflict
    - A team that works well together creates a **better product** for the client

Example 2: IT support company owner

- An **IT company owner** managing technicians who visit clients should:
  - Provide **fair pay** and ongoing **training** to keep staff skilled and motivated
    - Staff feel **valued**, which improves performance
  - Promote **ethical behaviour** with clients (e.g. don't overcharge or mislead)
    - Builds **trust** within the team and with clients, leading to **better service**
  - Create a culture of **respect and teamwork**, even under pressure
    - Helps staff support each other and deliver a **high-quality solution** to clients



Worked Example

Aisha manages a team of software developers.

Explain the reasons why it is important that Aisha acts ethically in relation to her team members. **[2]**

Answer

- To make sure the team members feel valued [1 mark]

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- To get the best work out of the team [1 mark]
- To enable them to work well together [1 mark]
- To enable them to create the best product for the client [1 mark]



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Copyright

What is the Copyright Designs & Patents Act?

- This protects the **intellectual property** of an individual or a company
- It makes it **illegal** to copy, modify or distribute software or other intellectual property without the relevant **permission**
- If **original work** is original, copyright will be **automatically applied** and will not expire until 25 - 70 years from the death of the creator depending on the type of work
- If an individual believes that their work has been copied it is their responsibility to take action under the **Copyright Designs and Patents Act**
- Many sites online offer free downloads of copyrighted **software/videos** which prevents the intellectual copyright holder from earning their income on the work they have created
  - E.g. If someone downloaded videos from Netflix and shared them with others, they would be breaching the act
- The act covers videos and audio where **peer-to-peer streaming** prevents a copyright owner from receiving an income

What is Prohibited Under the Copyright, Designs & Patents Act?

Primary breaches:

- **Copying** an original work
- **Issuing** the copy of the original work to the public
- **Renting/lending** the copy of the original work to the public
- **Performing**, **showing** or **playing** the original work in public
- **Making** an adaptation of the original work

Secondary breaches:

- **Importing** a copy of original work
- **Possessing** or dealing with a copy of the original work
- **Providing** means to make copies of the original work
- **Permitting** the use of premises for making copies of the original work
- **Provision** of props/equipment for a performance of a copy of the original work

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Case Study

Case Study

Ed Sheeran Vs Marvin Gaye (External link to bbc.co.uk)

- A case was brought against Ed Sheeran about his song 'Thinking Out Loud'
- Ed Sheeran won his case which ruled that he did not copy 'Let's Get It On' by Marvin Gaye whilst composing 'Thinking Out Loud'
- The heirs of Gaye's co-writer argued that Sheeran owed them money for infringement of copyright
- Sheeran said that if he was found guilty he would give up his music career
- The songs were said to have a similar chord progression but it was argued that these are the base of all modern songs and should be free to use
- Sheeran won a high court battle in London in 2022 over the copyright of his 2017 song 'Shape of You'
- In 2015 Gaye's heirs won a $5.3m judgement from a lawsuit claiming that Robin Thicke's 'Blurred Lines' copied Gaye's 'Got to Give it Up'

Your notes

Examiner Tips and Tricks

Examiner Tips and Tricks

- Make sure that you specify that the Copyright Design and Patents Act covers original work and is automatically applied from the creation of the work, then talk about what this means for the distribution of the work. For example:
  - *The work was protected by the Copyright Designs and Patents act when it was created. Therefore if anyone wishes to distribute it they have to gain the owner's permission, if they just posted it to the internet then this would be in breach of the act.*
- Also when referring to the name of the act, ensure you use its full name (The Copyright, Designs and Patents Act). It's not called the Copyright Act. If you use a shortened version it may cost you marks in the exam

Software licensing

Commercial

- Users must **pay** for the software
- The developer earns an **income** from each purchase
- The software is **copyrighted**, so:
  - Users **cannot legally edit, copy, or resell** the program
  - The developer **retains full control** over the product

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- Legal action can be taken against **piracy or illegal distribution**



Your notes

Shareware

- Software is **free to try**, often with time or feature limitations
- Users must **pay** to unlock the **full version**
- The software is **copyrighted**, so:
  - Users **cannot edit or redistribute** the program legally
  - The developer **retains control**
- This model encourages users to try before buying, increasing the chance of a sale

Freeware

- Software is **free to use and download**
- It is still **copyrighted**:
  - Users **cannot legally edit, copy, or sell** the software
  - The developer **keeps ownership and control**
- Although free, users must still follow the licence conditions

Open Source Initiative (OSI)

- The Open Source Initiative promotes software that is **freely available and open to everyone**
- **Open source software** comes with a licence that:
  - Allows users to **view, edit, and share** the source code
  - Encourages **collaboration and improvement** by the community
- Open source does **not always mean free**, some open-source software is paid, but users can still see and change the code
- Developers must still **credit the original authors** when modifying or sharing the software
- Example: **Linux**, **Firefox**, **Apache Web Server**

Free Software Foundation (FSF)

- The Free Software Foundation promotes **freedom**, not just free price
- It supports software that gives users four essential freedoms:
  1. **Freedom to run** the program for any purpose
  2. **Freedom to study** how the program works, and change it
  3. **Freedom to redistribute** copies
  4. **Freedom to improve** the program and share improvements

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- This type of software is often referred to as **free software** or **libre software**, and it usually provides access to the **source code**
- The FSF supports **user rights**, **transparency**, and **collaboration**
- Example: **GNU/Linux**, **LibreOffice**, **GIMP**

Feature	Commercial	Shareware	Freeware	Open Source (OSI)	Free Software (FSF)
Cost to use	Must be paid for	Free trial, then pay for full version	Free to use	Often free, but not always	Always free to use
Access to source code	No	No	No	Yes	Yes
Can be edited or modified	No – protected by copyright	No – protected by copyright	No – cannot edit or change	Yes – with conditions under OSI licences	Yes – full freedom to modify
Redistribution allowed	No	No	No	Yes (under licence terms)	Yes (freedom to share and improve)
Copyrighted	Yes	Yes	Yes	Yes	Yes
Developer control	Developer retains full control	Developer retains control	Developer retains control	Shared control between developers and users	Users have full control
Focus	Developer income and protection	Try before you buy	Free use only	Collaboration and transparency	Software freedom and user rights



Worked Example

A game will be released under a licence.

Tick one or more boxes in each row to identify the licence(s) each statement describes. **[4]**



Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

Statement	Free Software Foundation	Open Source Initiative	Shareware	Commercial Software
The user can edit the source code				
The user must always pay before being able to use the software				
The user can redistribute the software				
The user always gets a trial period				

Answer

1 mark for each correct column

Statement	Free Software Foundation	Open Source Initiative	Shareware	Commercial Software
The user can edit the source code	✓	✓		
The user must always pay before being able to use the software				✓
The user can redistribute the software	✓	✓		
The user always gets a trial period			✓	



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Understanding AI

What is artificial intelligence?

- Artificial intelligence (**AI**) is a machine that can **simulate intelligent behaviours** similar to that of a **human**
- AI is a system that can:
  - **Learn** - acquire new information
  - **Decide** - analyse and make choices
  - **Act autonomously** - take actions without human input
- There are two main types of AI:
  - **Weak AI**, also known as **narrow AI**, is designed to perform a specific task or set of tasks
  - **Strong AI**, also known as artificial general intelligence (**AGI**), is designed to perform any intellectual task that a human can do

Advantages and disadvantages of AI

Advantages	Disadvantages
Increased efficiency	Job losses
Increased accuracy	Potential for biased decision making
Scalability	Ethical concerns over its use

Characteristics of AI

- AI shares **three** common characteristics:
  - **Collection of data**
  - **Rules for using data**
  - **Ability to reason**

Collection of data	Rules for using data	Ability to reason
AI systems require **large amounts of data** to perform tasks	AI systems can use **logical reasoning** to evaluate information	It can **change its own rules**

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

The data is processed using **rules or algorithms** that enable the system to make decisions and predictions	and **make decisions** based on that information	and **data**



Your notes

Impact of AI

Social

Workforce

- AI adoption is expected to significantly **change employment structures**:
  - **Automation** may replace some roles, leading to **unemployment** or **job role changes**
  - At the same time, **new jobs will emerge** that require AI knowledge or human–AI collaboration
  - To support this shift, **reskilling and upskilling programmes** are essential, ensuring the workforce is prepared for AI-driven transformations

Education & Accessibility

- The rise of AI introduces concerns around **equal access**:
  - Those with better **technology, education, and internet access** may benefit more from AI, creating a **digital divide**
  - To prevent **social disparities**, it is important to ensure **equal access to AI education, tools, and training** for all communities

Healthcare

- AI has the potential to transform healthcare, but raises important **ethical and safety concerns**:
  - AI can improve **diagnosis, treatment planning**, and **patient monitoring**
  - However, AI systems are **not infallible** – a wrong diagnosis or treatment recommendation can have **serious consequences**
  - It is vital to determine the **extent of human oversight**
  - Decisions made **solely by AI** in critical situations can be risky
  - The question of **who is responsible** when something goes wrong (the developer, the AI system, or the healthcare provider) creates **complex legal and ethical challenges**
  - Clear **guidelines and regulations** are needed to define responsibility and ensure patient safety

Economic

Study copy distributed by EduResource ZA | eduresourcza.com

Employment & industry



Your notes

- AI adoption can significantly reshape entire industries, leading to:
  - **Job displacement** in sectors that rely heavily on routine or manual work
  - **Increased productivity** and efficiency in areas such as manufacturing, logistics, and finance
  - A growing demand for **AI-related roles**, such as data scientists and machine learning engineers
- To minimise negative effects, **governments and businesses must invest in retraining and upskilling programmes** to help workers transition into new roles

Business & innovation

- AI can be a catalyst for **economic growth** by:
  - Enabling **new business models**, such as personalised services or automated customer support
  - Supporting **faster innovation cycles** by improving R&D processes.
  - Reducing operational costs through automation and predictive analytics
- However, **small businesses may struggle** to compete with large companies that have more resources to invest in AI
- Potentially widening **economic inequalities** between organisations

Market dynamics & inequality

- The deployment of AI can:
  - **Concentrate wealth and power** in large tech companies that control key AI tools and data
  - Create **monopolistic advantages**, leading to reduced market competition
  - Require policymakers to consider **new economic models** and **regulation** to ensure fair access to AI technologies and prevent deepening income inequality

Environmental

Energy consumption

- AI systems, especially large-scale models, require vast computing power, which can:
  - Lead to **high electricity usage** and significant **carbon emissions**
  - **Strain power grids** if deployed at scale without renewable energy support
- Efforts must be made to:
  - Optimise AI models to be more **energy-efficient**
  - Encourage the use of **green data centres** powered by **renewable energy** sources

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Climate modelling & sustainability



Your notes

- AI can be a **powerful tool for environmental protection**:
  - Helps in **climate modelling**, predicting weather patterns and analysing environmental data
  - Aids in **optimising energy use**, improving efficiency in smart grids and buildings
  - Supports **sustainable agriculture** by analysing soil, weather, and crop data to reduce waste and overuse of resources
- However, the **positive environmental applications of AI must be weighed against its resource demands**, ensuring that the net impact supports climate goals

E-waste and hardware lifespan

- AI adoption drives demand for **specialised hardware** (e.g. GPUs, TPUs), which:
  - Can shorten the **lifespan of devices** due to rapid advancements in AI capability
  - Increases the volume of **electronic waste**, adding pressure to recycling systems
- Sustainable practices in **hardware design**, **recycling**, and **component reuse** are essential to reduce AI's environmental footprint



Worked Example

Aisha manages a team of software developers.

The team are developing a computer game where the user plays a board game (such as chess) against the computer.

Describe how the computer would use Artificial Intelligence (AI) to play the board game. **[3]**

Answer

- The rules / past moves / decision making algorithms of the game will be stored [1 mark]
- The AI program is trained, by playing many times [1 mark]
- AI will look (ahead) at possible moves [1 mark]
- … and/or analyse the pattern of past choices [1 mark]
- … and choose the move most likely to be successful [1 mark]
- Computer could learn how to improve // learn from previous mistakes [1 mark]
- … by storing the positive/negative result of choices [1 mark]
- … and changing its future choices [1 mark]

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Database Concepts

Contents

- Database models
- Database design
- Database normalisation

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Limitations of file-based systems

What is a file-based system?

- A file-based system refers to any system where **data is stored in separate files** (e.g. text files, spreadsheets)
- Each application or department may have its **own files**, often with **redundant or duplicated data**
- There is **no central control** or shared structure, files are often standalone
- Common in older systems or small-scale setups

Limitations

Limitation	Explanation
Data redundancy	The same data is stored in multiple files, leading to duplication
Data inconsistency	If one file is updated but others aren't, the data becomes unreliable
Lack of data integrity	It's harder to ensure data is accurate and consistent across all files
Poor security	Limited control over who can access or modify different files
Difficult to update	Changes must be made in several places, increasing time and risk of error
Limited scalability	Not suitable for large volumes of data or complex data relationships
No central control	Each application manages its own data, leading to lack of coordination
Hard to manage relationships	Cannot easily link related data across files (e.g. customers and their orders)

Flat file vs Relational

- A **flat file database** is one that stores all data in a **single table**

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- It is simple and easy to understand but causes data redundancy, inefficient storage and is harder to maintain

- A **relational database** is one that organises data into **multiple tables**
  - It uses keys to connect related data which reduces data **redundancy**, makes efficient use of storage and is easier to maintain

- Consider this example **flat file** table of students

ID	first_name	last_name	Personal tutor	FormRoom
1	sam	smith	Roger Hinds	6b
2	fred	lynch	Jess Little	8j
3	depak	noor	Roger Hinds	6b
4	archie	henns	Mary Kent	8k
5	helga	jordan	Mary Kent	8k
6	lizzy	bell	Mary Kent	8k
7	xavier	horten	Jack Berry	3m

- This table has **redundant data** - the tutor and form room information repeats
- This is **inefficient**
- If a tutor changed their name we would need to find all instances of that name and change them all
- Missing any would mean the table had **inconsistent data**
- A **relational** database would solve this issue:
  - A new **table** could be created to store the tutor information and the tutor information in the student table could be moved to the new **table**
  - A **foreign key** in the student **table** (TutorID) could link a student to their tutor



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com



Your notes

ID	first_name	last_name	TutorID
1	sam	smith	1
2	fred	lynch	2
3	depak	noor	1
4	archie	henns	3
5	helga	jordan	3
6	lizzy	bell	3
7	xavier	horten	4

TutorID	Tutor Name	FormRoom
1	Roger Hinds	6b
2	Jess Little	8j
3	Mary Kent	8k
4	Jack Berry	3m

- Now the name of each tutor and their form room is stored **only once**
- This means if they change only one piece of data, the data is **updated in the entire database** and **inconsistency** is avoided

Database terminology

Term	Definition
Entity	A real-world object or concept that data is stored about (e.g. Student, Book)
Table	A collection of data about an entity, organised in rows and columns
Record (Tuple)	A single row in a table representing one instance of an entity
Field (Attribute)	A single column in a table, storing one piece of data about the entity

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Primary key	A unique identifier for each record in a table (e.g. StudentID)
Candidate key	A field (or combination of fields) that could be used as a primary key
Secondary key	A field used for searching or sorting, but not necessarily unique
Foreign key	A field that links to the primary key in another table to create relationships
Relationship	A logical connection between two tables/entities
One-to-One	One record in a table relates to one record in another table
One-to-Many	One record in a table relates to many records in another table
Many-to-Many	Records in one table relate to many records in another, and vice versa
Referential integrity	Ensures foreign keys match a primary key in the related table to prevent broken links
Indexing	A technique to speed up searching in a table by creating an ordered list of key fields



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Entity-relationship (E-R) diagrams

What is an entity?

- An entity in something worthy of **capturing and storing data about** e.g. *students, orders, products, courses, customers*
- Entities become **tables** in a **relational database**
- Relational databases store different entities in **separate tables**
- Linking tables depends on the **relationships between entities**
- There are 3 types of (sometimes called degrees of) **relationships:**
  - **One-to-one**
  - **One-to-many**
  - **Many-to-many**
- Imagine a company has
  - A table of products
  - A table of customers
  - A table of the orders the customers have made
- What is the relationship between a **customer** and an **order**?
  - **One** customer can make multiple (**many**) orders
  - But each order relates to a specific (**one**) customer
  - So the relationship between customer and order is **one-to-many**
- Now consider the relationship between a **product** and an **order**
  - An order could have more than one (**many**) products on it
  - A product could be on more than one (**many**) order
  - So the relationship between order and product is **many-to-many**
- **One-to-one** relationships also exist but are not very common in databases

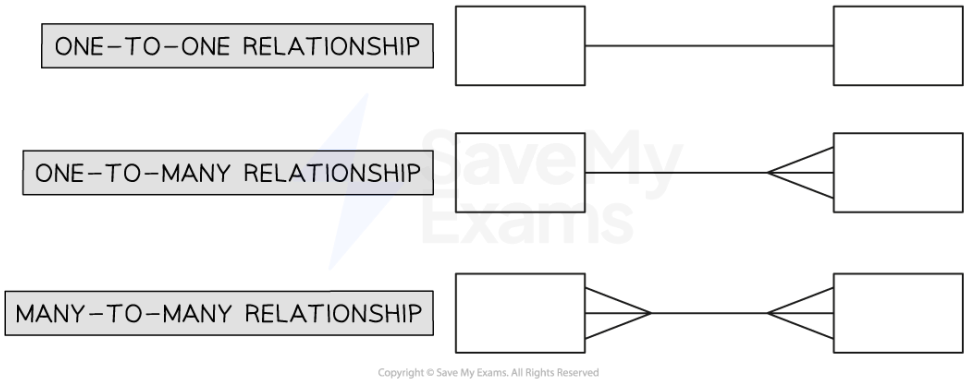
What is an entity-relationship (E-R) diagram?

- An entity relationship diagram (**E-R**) is a **diagram that represents the entities** (tables) that will be in a database and the **relationships between these entities**
- The entities are drawn as **boxes** with the **entity** name in
- The relationships are drawn in as what is known as '**crow's feet notation**'

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- This is how to draw the relationships in the exam:



Copyright © Save My Exams. All Rights Reserved

- The **names** of the entities would go inside the boxes

Examiner Tips and Tricks

These diagrams are simple but tell us some important things about the database:

- The names of all the tables
- Which tables will have a foreign key - when an entity has a 'many' relationship against it that means it will have a foreign key in it that links to the primary key of the connected entity

Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Normalised design

What is a normalised design?

- A **normalised design** is a database structure that:
  - **Organises data efficiently**
  - **Eliminates unnecessary duplication**
  - **Ensures data integrity**
  - **Supports easy updates and accurate relationships**
- You may be asked to produce a normalised design based on:
  - A **written description** of a system
  - A **table of unnormalised data**
  - A set of **existing flat tables**
- You need to be able to:
  - Identify **repeating groups** or **duplicated data**
  - Break down data into **separate related tables**
  - Assign appropriate **primary keys** and **foreign keys**
  - Show the database in **1NF, 2NF, and 3NF**
  - Clearly show **relationships** between the tables

Step-by-step process

Step	What to do	Why it matters
1	Read the scenario or table carefully	Understand what entities (real-world things) are involved
2	Identify the fields and any repeated/duplicated values	These usually indicate the need for more than one table
3	Apply **1NF** – remove repeating groups, separate fields	Each field should hold a single value
4	Apply **2NF** – remove partial dependencies	Make sure non-key fields depend on the whole primary key

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

5	Apply **3NF** – remove transitive dependencies	Non-key fields should not depend on other non-key fields
6	Assign **primary keys** for each table	Every table needs a unique identifier
7	Use **foreign keys** to link related tables	Ensures relationships and referential integrity



Your notes

Example

- A student is enrolled on **multiple courses**
- Each course is taught by **a teacher**
- The table includes: StudentName, StudentID, CourseID, CourseName, TeacherName
- You would:
  - Identify **Entities**: Student, Course, Teacher
  - Create tables:
    - **Student** (StudentID, StudentName)
    - **Course** (CourseID, CourseName, TeacherID)
    - **Teacher** (TeacherID, TeacherName)
    - A link table: **StudentCourse** (StudentID, CourseID) – handles many-to-many



Examiner Tips and Tricks

- Always check for **repeating fields** or **duplicated values**
- Watch for **composite keys** in link tables (especially many-to-many relationships)
- Check that all fields in a table **depend only on the key, the whole key, and nothing but the key**
- Label each stage of your work (1NF → 2NF → 3NF)

First Normal Form (1NF)

What is first normal form?

- For a **table** to be in first normal form it must:
  - **Contain atomic values**
    - Each column in a table must contain single, indivisible values
  - **Have no repeating groups**
    - Columns must not contain arrays or lists of values

Study copy distributed by EduResource ZA | eduresourcza.com



Your notes

- **Have unique column names**
  - Each column must have a unique name within the table
- **Have a unique identifier (primary key)**
  - Each row must have a unique identifier to distinguish it from other rows

- This customers table below has **no primary key** and the name is stored in **one field** so is not atomic
- This table is **not** in first normal form

Table: Customers

name	phone	country
John Smith	07373 929122	UK
Iram Iravani	07234 543422	Iraq
Wu Zhang	04563 523427	China
Anne James	09378 482894	USA
Khalid Shirvani	02343 536522	France

- This customers table below **has a primary key** and the name is stored in **two fields** so it is **atomic**
- This table is in **first normal form**

Table: Customers

customer_id	forename	surname	phone	country
1	John	Smith	07373 929122	UK
2	Iram	Iravani	07234 543422	Iraq
3	Wu	Zhang	04563 523427	China
4	Anne	James	09378 482894	USA
5	Khalid	Shirvani	02343 536522	France

Second Normal Form (2NF)

What is second normal form?

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com



Your notes

- For a table to be in second normal form it must:
  - **Fulfil all 1NF requirements**
  - **Only apply to tables with a compound primary key**
  - **Have full functional dependency**
    - **All non-prime attributes (attributes not part of the primary key) must be fully dependent on the primary key**
  - **Have no partial dependencies**
    - Non-prime attributes must not depend on only part of the primary key (in case of a composite primary key)
    - Separate tables should be created for partially dependent attributes
- In this table below, Course Title **only depends on part** of the compound primary key (the course code) and not the Date so this table **is not** in second normal form

Table: Course

Course	Date	Course Title	Room	Capacity	Available
SQL101	03/01/2020	SQL Basics	4A	12	4
DB202	03/01/2020	Database Design	7B	14	7
SQL101	04/05/2020	SQL Basics	7B	14	10
SQL101	15/05/2020	SQL Basics	12A	8	8
CS50	31/05/2020	C Programming	4A	12	11

- To turn this table into second normal form we will ensure:
  - Course Title is moved into its own **Course** table
  - A Session table is created to use the full key (Course, Date) for all time-specific info
  - No **partial dependencies** remain

Table: Course

Course	Course Title
SQL101	SQL Basics
DB202	Database Design

Study copy distributed by EduResource ZA | eduresourcza.com

CS50	C Programming

Table: Session

Course	Date	Room	Capacity	Available
SQL101	03/01/2020	4A	12	4
DB202	03/01/2020	7B	14	7
SQL101	04/05/2020	7B	14	10
SQL101	15/05/2020	12A	8	8
CS50	31/05/2020	4A	12	11

Third Normal Form (3NF)

What is third normal form?

- For a table to be in third normal form it must:
  - **Fulfil all 2NF requirements**
  - **Have no transitive dependencies**
    - Non-prime attributes must not depend on other non-prime attributes
  - **Have each non-prime attribute dependent solely on the primary key, not on other non-prime attributes**
  - **Have separate tables for attributes with transitive dependencies, and the tables should be linked using a foreign key**
- In this table below, the certificate depends on the title - this a transitive dependency and so this table **is not** in third normal form

FilmID	Title	Certificate	Description
12034	Saw IV	18	Eighteen and over
12035	Spiderman 2	12A	Age 12 and over
12036	Shrek	U	Universal

- To turn this table into third normal form we will ensure:

Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- Transitive dependency removed by **separating** Description into its own table
- The Film table only stores fields that **directly depend** on the key (FilmID)
- A **foreign key** (Certificate) is used to maintain the relationship



Your notes

Table: Film

FilmID	Title	Certificate
12034	Saw IV	18
12035	Spiderman 2	12A
12036	Shrek	U

Table: Certificate

Certificate	Description
18	Eighteen and over
12A	Age 12 and over
U	Universal



Worked Example

Match each normal form to its definition:

First Normal Form (1NF)	All fields are fully dependent on the primary key.
Second Normal Form (2NF)	There are no repeating groups of attributes.
Third Normal Form (3NF)	There are no partial dependencies.

[1]

Answer

First Normal Form (1NF)	There are no repeating groups of attributes.
Second Normal Form (2NF)	There are no partial dependencies.
Third Normal Form (3NF)	All fields are fully dependent on the primary key.

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Database Management Systems (DBMS)

Contents

- DBMS features

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

DBMS vs file-based systems

What is a DBMS?

- A DBMS (Database Management System) is **software** that is used to **create, manage, and interact with databases**
- A DBMS provides several **key features**, including:
  - Defines a **logical schema** – the overall logical design of the database (tables, fields, relationships, and constraints), showing how data is organised independently of how it is physically stored
  - Provides **data security** and **access control** to keep data safe
  - Maintains **data integrity** by enforcing rules and constraints, e.g.
    - If the data is changed in one place it is updated in every other place
    - Ensures each foreign key has a corresponding primary key
  - Supports **concurrency control** so multiple users can access the database at the same time without errors
  - Ensures **data independence** so changes to storage methods do not affect how data is used
  - Offers **backup and recovery** tools to protect against data loss
- It addresses the limitations of file-based systems by:

Area	Limitation in file-based systems	How a DBMS solves it
Data redundancy	Same data is stored in multiple files, wasting space and causing confusion	A DBMS **centralises data**, so it is stored once and referenced when needed
Data inconsistency	Data may differ between files if one is updated but others are not	A DBMS ensures **data consistency** by updating a single source of truth
Data management	Each application stores and manages its own data	A DBMS provides a **data dictionary** that defines all data elements across the system (field/table names, data types)

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Data modelling	No structured way to represent or link data	A DBMS uses **data modelling tools** (e.g. ER diagrams) to define entities and relationships
Logical schema	The overall logical design of the database (tables, fields, relationships, and constraints)	Shows how data is **organised and used**, independently of how it is physically stored
Data integrity	Difficult to enforce validation rules across different files	A DBMS uses **constraints** (e.g. primary keys, foreign keys) to enforce integrity
Data security	Poor control over who can access or change data	A DBMS supports **user accounts, access rights, and permissions** for individuals/groups
Backup procedures	Often manual and inconsistent	A DBMS includes **automated backup and recovery tools** to reduce risk of data loss



Your notes

DBMS tools

Developer interface

- The developer interface allows developers to **write database queries** using **Structured Query Language (SQL)**
- This is more flexible than using **query-by-example** tools, which are visual and simplified
- Using SQL allows the developer to create **more complex and precise queries** to search, update, or manage data

Query processor

- The query processor is responsible for **handling and executing SQL queries**
- It has several key components:

Component	Function
DDL interpreter	Interprets **Data Definition Language (DDL)** commands like `CREATE`, `ALTER`, and `DROP` and updates the **data dictionary**
DML compiler	Compiles **Data Manipulation Language (DML)** statements (e.g. `SELECT`, `INSERT`, `UPDATE`) into low-level instructions and **optimises** the query

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Query evaluation engine	Executes the compiled instructions to **retrieve or manipulate** the actual data in the database



Your notes

- Together, these components make sure that any query written by the developer is **interpreted, optimised**, and **executed efficiently**



Worked Example

State what is meant by a data dictionary and give one example of an item typically found in a data dictionary. **[2]**

Answer

- A data dictionary is data about the data structure in the database [1 mark]
- Example: field name, data type, or table name [1 mark]

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Data Definition Language (DDL) & Data Manipulation Language (DML)

Contents

- SQL usage

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

DDL & DML

What is DDL & DML?

- Data Definition Language (**DDL**) is used to **build and manage the structure** of a relational database
- It defines **how data is stored** by creating and modifying tables, fields, indexes, and relationships
- Data Manipulation Language (**DML**) is used to **manage the data** stored within the database structures created by DDL
- It deals with **adding, updating, deleting**, and **retrieving** data
- Both DDL and DML use **SQL syntax** to interact with the database
- DDL **structures** the database, while DML maintains the **contents** of the database
- Most modern **DBMSs use SQL** to support both DDL and DML operations

Basic DDL commands

Command	Purpose	Example
CREATE DATABASE	Creates a new database	CREATE DATABASE SchoolDB
CREATE TABLE	Creates a new table with specified fields and data types	CREATE TABLE Students (StudentID INTEGER, Name VARCHAR(50))
Data types for attributes:	Defines the type of data each field can store	
CHARACTER	Fixed-length text	Name CHARACTER(10)
VARCHAR(n)	Variable-length text (max length n)	Email VARCHAR(100)
BOOLEAN	True or False value	IsActive BOOLEAN
INTEGER	Whole numbers	Age INTEGER
REAL	Decimal numbers	Score REAL

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

DATE	Calendar date	DateOfBirth DATE
TIME	Time of day	StartTime TIME
ALTER TABLE	Changes an existing table (e.g. add or remove fields)	ALTER TABLE Students ADD Email VARCHAR(100)
PRIMARY KEY (field)	Sets a field as the primary key (unique identifier)	PRIMARY KEY (StudentID) inside a CREATE TABLE or ALTER TABLE command
FOREIGN KEY (field) REFERENCES Table(Field)	Sets up a foreign key relationship between tables	FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)



Your notes

Example: School database

- You are designing a database to store information about **students** and the **courses** they are enrolled in
- Each student can enrol in many courses, and each course can have many students (a **many-to-many relationship**)
- Three tables are needed:
  - Students – to store student details
  - Courses – to store course details
  - Enrolments – a link table to handle the many-to-many relationship

```
-- Create the Students table
CREATE TABLE Students (
    StudentID INTEGER,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    DateOfBirth DATE,
    PRIMARY KEY (StudentID)
);

-- Create the Courses table
CREATE TABLE Courses (
    CourseID INTEGER,
    CourseName VARCHAR(100),
    Credits INTEGER,
    PRIMARY KEY (CourseID)
);

-- Create the Enrolments table to link Students and Courses
CREATE TABLE Enrolments (
    StudentID INTEGER,
    CourseID INTEGER,
    EnrolDate DATE,
    PRIMARY KEY (StudentID, CourseID),
    FOREIGN KEY (StudentID) REFERENCES Students(StudentID),
```


Study copy distributed by EduResource ZA | eduresourcza.com

```
    FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)
);
```



Your notes

Basic DML commands

Retrieving data

Command	Purpose	Example
SELECT ... FROM	Retrieves specific columns from a table	SELECT FirstName, LastName FROM Students
WHERE	Filters results based on a condition	SELECT * FROM Students WHERE Age > 16
ORDER BY	Sorts the results in ascending or descending order	SELECT * FROM Students ORDER BY LastName ASC
GROUP BY	Groups rows that have the same value in specified columns	SELECT CourseID, COUNT(*) FROM Enrolments GROUP BY CourseID
INNER JOIN	Combines rows from two tables where there is a matching key	SELECT Students.FirstName, Courses.CourseName FROM Students INNER JOIN Enrolments ON Students.StudentID = Enrolments.StudentID INNER JOIN Courses ON Enrolments.CourseID = Courses.CourseID
SUM()	Calculates the total of a numeric column	SELECT SUM(Credits) FROM Courses
COUNT()	Counts the number of rows (or non-null entries in a column)	SELECT COUNT(*) FROM Enrolments
AVG()	Calculates the average value of a numeric column	SELECT AVG(Age) FROM Students

Data maintenance

Command	Purpose	Example
INSERT INTO	Adds a new record to a table	INSERT INTO Students (StudentID, FirstName, LastName) VALUES (1, 'Ali', 'Khan')

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

DELETE FROM	Removes records from a table	DELETE FROM Students WHERE StudentID = 1
UPDATE	Modifies existing data in a table	UPDATE Students SET Age = 17 WHERE StudentID = 1



Your notes

Example: School database

- You're working with a school database containing three tables:
  - Students(StudentID, FirstName, LastName, Age)
  - Courses(CourseID, CourseName, Credits)
  - Enrolments(StudentID, CourseID, EnrolDate)
- You want to **query the database** and **maintain the data** as students enrol in courses, update their details, or leave the school

Querying

```
-- 1. Get the first and last names of all students
SELECT FirstName, LastName FROM Students;

-- 2. List students who are older than 16
SELECT * FROM Students WHERE Age > 16;

-- 3. List all students sorted alphabetically by surname
SELECT * FROM Students ORDER BY LastName ASC;

-- 4. Count how many students are enrolled on each course
SELECT CourseID, COUNT(*) FROM Enrolments GROUP BY CourseID;

-- 5. Show student names with the courses they are enrolled on
SELECT Students.FirstName, Students.LastName, Courses.CourseName
FROM Students
INNER JOIN Enrolments ON Students.StudentID = Enrolments.StudentID
INNER JOIN Courses ON Enrolments.CourseID = Courses.CourseID;

-- 6. Get the total number of course credits available
SELECT SUM(Credits) FROM Courses;

-- 7. Get the average age of all students
SELECT AVG(Age) FROM Students;
```

Maintaining

```
-- 1. Add a new student to the Students table
INSERT INTO Students (StudentID, FirstName, LastName, Age)
VALUES (4, 'Amira', 'Patel', 17);

-- 2. Enrol that student on CourseID 2
INSERT INTO Enrolments (StudentID, CourseID, EnrolDate)
VALUES (4, 2, '01/09/2025'); -- UK date format (DD/MM/YYYY)

-- 3. Update a student's age
UPDATE Students SET Age = 18 WHERE StudentID = 4;
```


© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

```
-- 4. Delete a student who has left the school
DELETE FROM Students WHERE StudentID = 4;
```

Your notes

Worked Example

Part of a database table BIRD_TYPE is shown:

BirdID	Name	Size
0123	Blackbird	Medium
0035	Jay	Large
0004	Raven	Large
0085	Robin	Small

The database only supports these data types:

- character
- varchar
- Boolean
- integer
- real
- date
- time

Write a Structured Query Language (SQL) script to define the table Bird_Type. **[4]**

Answer

- CREATE TABLE start and end bracket [1 mark]
- Bird ID as CHAR/VARCHAR [1 mark]
- Name **and** size as VARCHAR/CHAR [1 mark]
- Bird ID as primary key [1 mark]

Example answer

```
CREATE TABLE BIRD_TYPE(
BirdID CHAR(4) NOT NULL,
Name VARCHAR(9),
Size VARCHAR(6),
PRIMARY KEY (BirdID)
);
```


Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Computational Thinking Skills

Contents

- Computational Thinking Skills

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Abstraction

What is abstraction?

- Abstraction is the process of **removing unnecessary details** from a problem to **focus on the important features** for implementing a solution
- Examples of abstraction include modelling a:
  - **real-life object**
  - **environment**
  - **action**
  - **sequence of actions**
  - **concept**
- Implementations of these include:
  - **a computer game that simulates playing a sport**
  - **a simulator such as a car or flight simulator,**
  - **a map of a bus or train route in a city**
- When creating a program, developers must **identify important features** that will contribute to solving the problem or have a role to play in the solution

Computer games



© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- Computer games use a large amount of abstraction, **removing the elements that a user does not need to consider** in order to enjoy playing the game
- When using abstraction in computer games which are designed to simulate real life, the aim is to make the game realistic and visually appealing whilst keeping the game fun to play
- In a game that simulates a sport, it is important to the user that visually they recognise the environment and when they perform an action, they see a response
- However, users do not need to know the complex algorithms used to control the non player characters (NPCs)



Your notes

Train map

© 2026 Save My Exams, Ltd.

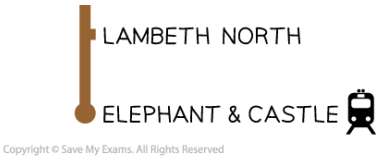
Study copy distributed by EduResource ZA | eduresourcza.com



- HARROW & WEALDSTONE
- KENTON
- SOUTH KENTON
- NORTH WEMBLEY
- WEMBLEY CENTRAL
- STONEBRIDGE PARK
- HARLESDON
- WILLESDON JUNCTION
- KENSAL GREEN
- QUEEN'S PARK
- KILBURN PARK
- MAIDA VALE
- WARWICK AVENUE
- PADDINGTON
- EDGEWARE ROAD
- MARYLEBONE
- BAKER STREET
- REGENT PARK
- OXFORD CIRCUS
- PICADILLY CIRCUS
- CHARING CROSS
- EMBANKMENT
- WATERLOO

Your notes

Study copy distributed by EduResource ZA | eduresourcza.com



Copyright © Save My Exams. All Rights Reserved

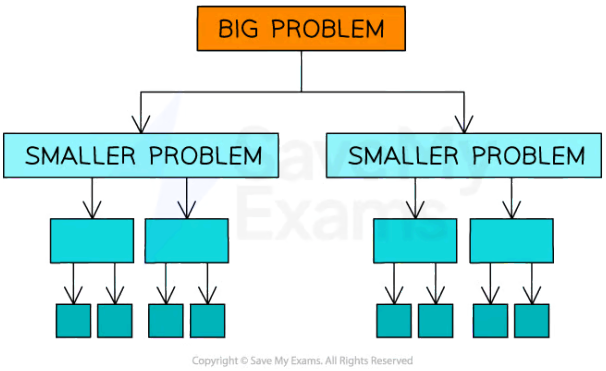
Your notes

- Another specific example of abstraction would be the London underground train route map; travellers **do not need to know the geographical layout** of the routes, only that getting on at stop A will eventually transport you to stop B

Decomposition

What is decomposition?

- Decomposition is **breaking down a big problem** into smaller problems so that they can be **solved independently**
- Programmers use decomposition to:
  - **Break** problems down
  - **Identify** the steps, parts or processes involved in a problem
  - **Identify** reusable components
  - **Split** tasks between programmers
- In the case of slow appointment booking, you could **decompose** the problem into issues such as server limitations during peak hours, inefficient backend algorithms, and user interface delays



Copyright © Save My Exams. All Rights Reserved

Process of decomposition to break problems down

Using abstraction to design a solution

- Before we start to break problems down, it is often useful to **apply the rules of abstraction** to the problem
- Applying abstraction will remove **non-essential elements** and that programmers can **focus on critical aspects** for problem-solving

Study copy distributed by EduResource ZA | eduresourcza.com

- When addressing a slow booking system, a programmer could ignore elements like the system's colour scheme or graphics and focus on critical performance metrics such as server response time and database query efficiency



Your notes

Decomposing slow response times problem

1. **Big problem**: Slow response times during the online appointment booking process
2. **Decompose** into sub-problems:
    - Server limitations causing **bottlenecks** during peak hours
    - Inefficient algorithms leading to **delayed confirmation** screens
    - Poor user interface contributing to an overall **bad user experience**
3. **Prioritise** sub-problems:
    1. Address **server limitations**, as solving this could have a broad impact on system performance
    2. Look into **optimising algorithms** to speed up the booking and confirmation process
    3. Lastly, make **user interface improvements** to enhance user experience, possibly mitigating some of the frustration caused by slow response times

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Algorithms

Contents

- Algorithm basics
- Pseudocode
- Translation skills
- Refinement & logic

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Definition of algorithm

What is an algorithm?

- An algorithm is a **solution to a problem** expressed as a **sequence of defined steps**
- Methods of writing an algorithm before attempting to program a solution include:
  - **Structured English**
  - **Pseudocode**
  - **Flowcharts**

Structured English

- Structured English is a **human-readable method** for describing algorithms using a combination of **natural English language** and **programming logic**
  - It uses clear English phrases to describe each step
  - Logic structures like **IF...THEN**, **REPEAT**, and **WHILE** may appear, but without strict syntax rules
  - Often used in the early planning stages before converting to pseudocode

```
Ask the user to enter their age
If the age is 18 or over
→ Display a welcome message
Otherwise
→ Display an access denied message
```

Pseudocode

- Pseudocode is a **precise, structured, and language-independent way** of describing an algorithm that resembles a programming language
- It **follows specific exam-board-defined syntax**
  - It includes formal elements like IF, THEN, ELSE, WHILE, REPEAT, DECLARE, and ← for assignment
  - Students must follow **CIE's pseudocode format** in exams

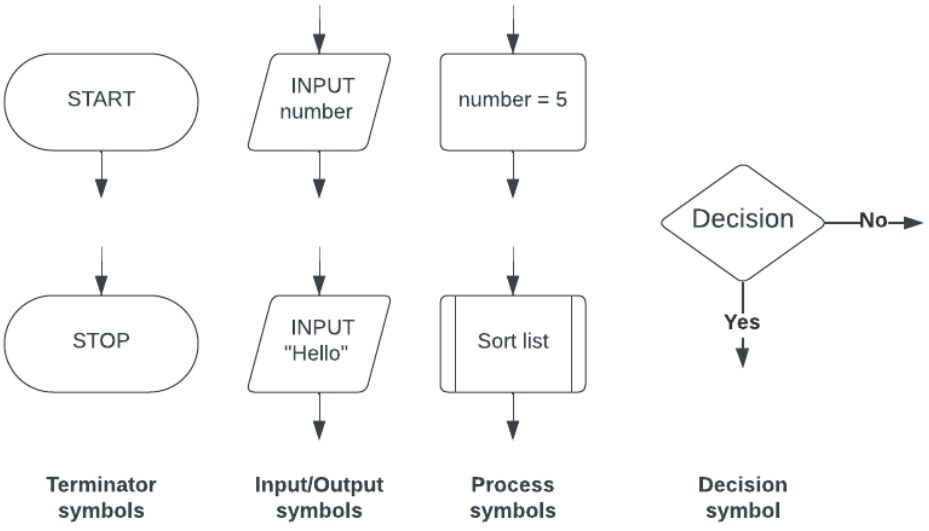
```
INPUT Age
IF Age >= 18 THEN
   OUTPUT "Welcome to the site"
ELSE
   OUTPUT "Sorry, this site is for users 18 and over"
ENDIF
```

Flowcharts

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- Flowcharts are a **visual tool** that uses **shapes to represent different functions** to **describe an algorithm**
- Used to **visualise the flow of control** in a system
- Standard symbols include:
  - **Oval** for Start/End
  - **Parallelogram** for Input/Output
  - **Rectangle** for Processes
  - **Diamond** for Decisions
- Arrows show the **sequence of operations**



Identifier tables

What is an identifier table?

- An identifier table is used when writing **pseudocode** to keep track of all the **identifier names** used in an algorithm
- An **identifier** is the **name given to a variable, constant, array, procedure**, or any other named element in the pseudocode

Why use an identifier table?

- It helps you stay **organised** when designing an algorithm
- Ensures **consistent naming**
- Makes it easier to understand what each identifier stores or does

Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- Useful in **exam questions** when you are asked to declare variables clearly



Your notes

Identifier naming rules

- Must start with a **letter** (A–Z or a–z)
- Can include **letters**, **digits (0–9)**, and **underscores (_)**
- **Accented characters** and **symbols** are not allowed
- Identifiers are **case sensitive** (e.g. Total and total are **NOT** treated the same)

Identifier	Description
StudentName	Stores a student's full name
TestScore	Holds a test score value
MAX_SCORE	Maximum score allowed
FormList	Stores names in a form group

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Input, process, output

What is an input?

- An input is **data or information being entered/taken into a program** before it is processed in the algorithm
- An input can come from a variety of sources, such as:
  - **User** - keyboard, mouse, controller, microphone
  - **Sensors** - temperature, pressure, movement
- Values are input using the INPUT command

```
INPUT <identifier>
```

What is a process?

- A process is **a doing action performed in the algorithm** that transforms inputs into the desired output. The central processing unit (**CPU**) executes the instructions that define the process
- An example would be:
  - **Comparing two numbers**
  - **Calculating an average**

What is an output?

- An output is **the result of the processing in an algorithm** and usually the way a user can see if an algorithm works as intended
- An output can take various forms, such as:
  - **Numbers** - result of calculations
  - **Text**
  - **Images**
  - **Actions** - triggering events
- Values are output using the OUTPUT command

```
OUTPUT <value(s)>
```

More than one value can be output in the same command when separated by commas

```
OUTPUT "First name: ", Fname, "Surname: ", Sname
```

Example 1 - Area of a shape

- A user wants to write a program to calculate the area of a shape

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Input	Process	Output
▪ Length ▪ Width	▪ Length X width	▪ Area


Your notes

```
// Calculate the area of a rectangle

INPUT Length
INPUT Width

Area ← Length * Width

OUTPUT "The area is ", Area
```

Example 2 – Average test score

- A teacher wants to calculate the average mark achieved on a test amongst students in a class
- The teacher needs to enter how many students in the class and for each students a score out of 50

Input	Process	Output
▪ Number of students ▪ Score per student	▪ TotalScore = TotalScore + score per student ▪ Average = TotalScore / Number of students	▪ Average mark

```
// Calculate the average test score for a class

DECLARE TotalScore : INTEGER
DECLARE Score : INTEGER
DECLARE NumberOfStudents : INTEGER
DECLARE Average : REAL

TotalScore ← 0

INPUT NumberOfStudents

FOR StudentIndex ← 1 TO NumberOfStudents
   OUTPUT "Enter score for student ", StudentIndex
   INPUT Score
   TotalScore ← TotalScore + Score
NEXT StudentIndex

Average ← TotalScore / NumberOfStudents

OUTPUT "The average score is ", Average
```

Programming constructs

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

What is sequence?



Your notes

- Sequence refers to lines of code which are run **one line at a time**
- The lines of code are run in the **order** that they written from the first line of code to the last line of code
- Sequence is **crucial to the flow** of a program, any instructions out of sequence can lead to **unexpected behaviour or errors**

Example

- A simple program to ask a user to input two numbers, number two is subtracted from number one and the result is outputted

Line	Pseudocode
01	INPUT NumberOne
02	INPUT NumberTwo
03	Result ← NumberOne - NumberTwo
04	OUTPUT "The result is ", Result

What is selection?

- Selection is when the **flow of a program is changed,** depending on a **set of conditions**
- The outcome of this condition will then determine **which lines or block of code is run next**
- Selection is used for **validation**, **calculation** and making sense of a **user's choices**
- There are two ways to write selection statements:
  - **if**... **then**... **else**...
  - **case**...

Structure	IF...THEN...ELSE example	CASE statement example
Purpose	Used for **binary decisions** (true/false)	Used for **multiple specific options**
Scenario	Check if a user is old enough to vote	Perform an action based on the direction entered

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Pseudocode example	INPUT Age IF Age >= 18 THEN OUTPUT "You can vote" ELSE OUTPUT "You cannot vote" ENDIF	INPUT Direction CASE OF Direction "N" : OUTPUT "North" "S" : OUTPUT "South" "E" : OUTPUT "East" "W" : OUTPUT "West" OTHERWISE : OUTPUT "Invalid direction" ENDCASE



Your notes

What is iteration?

- Iteration means **repeating a line or block of code** using a **loop**
- It allows a program to perform a task **multiple times** until a condition is met

Types of iteration

Type	Description	Pseudocode Format
Count-controlled	Repeats a fixed number of times	FOR ... TO ... NEXT
Post-condition	Repeats **at least once**, checks the condition **after** running the block	REPEAT ... UNTIL
Pre-condition	Checks the condition **before** running the block	WHILE ... ENDWHILE

```
// Count-controlled
FOR i ← 1 TO 5
  OUTPUT i
NEXT i

// Post-condition
REPEAT
  INPUT Password
UNTIL Password = "Secret123"

// Pre-condition
WHILE Score < 100
  Score ← Score + 10
ENDWHILE
```



Worked Example

A teacher uses a paper-based system to store marks for a class test. The teacher requires a program to assign grades based on these results.

The program will output the grades together with the average mark.

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Write a detailed description of the algorithm that will be needed. **[6]**

Answer

Marks awarded for a description of each of the following steps of the algorithm:

1. Reference variables for Count of students and Total marks [1 mark]
2. Loop through all students (Count) [1 mark]
3. Input individual mark (in loop) [1 mark]
4. Compare mark with threshold / boundary values to determine grade (in loop) [1 mark]
5. Output the grade for a student (in loop) [1 mark]
6. Maintain a Total (and Count if required) (in loop) [1 mark]
7. Calculate average by dividing Total by Count and Output (after loop) [1 mark]



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Structured English/Flowchart to Pseudocode

Example – Structured English

A customer is buying an item. If the customer has a discount code, they receive 10% off the price. Otherwise, they pay the full amount. The program should ask for the item price and whether they have a discount code, then calculate and display the final price.

Step 1 – Structured English

1. Ask the user to enter the price of the item
2. Ask the user if they have a discount code
3. If they do have a discount code
   → Calculate 10% off the price
   → Subtract the discount from the original price
4. Otherwise
   → The price stays the same
5. Output the final price

Step 2 – Identifier table

Identifier	Description
ItemPrice	Original price of the item
HasDiscount	TRUE if user has a discount code
DiscountAmount	Amount taken off the original price
FinalPrice	Price after discount is applied

Step 3 – Pseudocode

```
// Ask for item price
INPUT ItemPrice

// Ask if user has a discount code
INPUT HasDiscount

IF HasDiscount = TRUE THEN
   DiscountAmount ← ItemPrice * 0.10
   FinalPrice ← ItemPrice - DiscountAmount
ELSE
```


Study copy distributed by EduResource ZA | eduresourcza.com

```
    FinalPrice ← ItemPrice
ENDIF

// Output the final price
OUTPUT "The final price is ", FinalPrice
```



Your notes

Structured English/Pseudocode to Flowchart

Example – Pseudocode

```
INPUT Number

IF Number MOD 2 = 0 THEN
    OUTPUT "Even"
ELSE
    OUTPUT "Odd"
ENDIF
```

Step 1 – Identifier table

Identifier	Description
Number	The number entered by the user

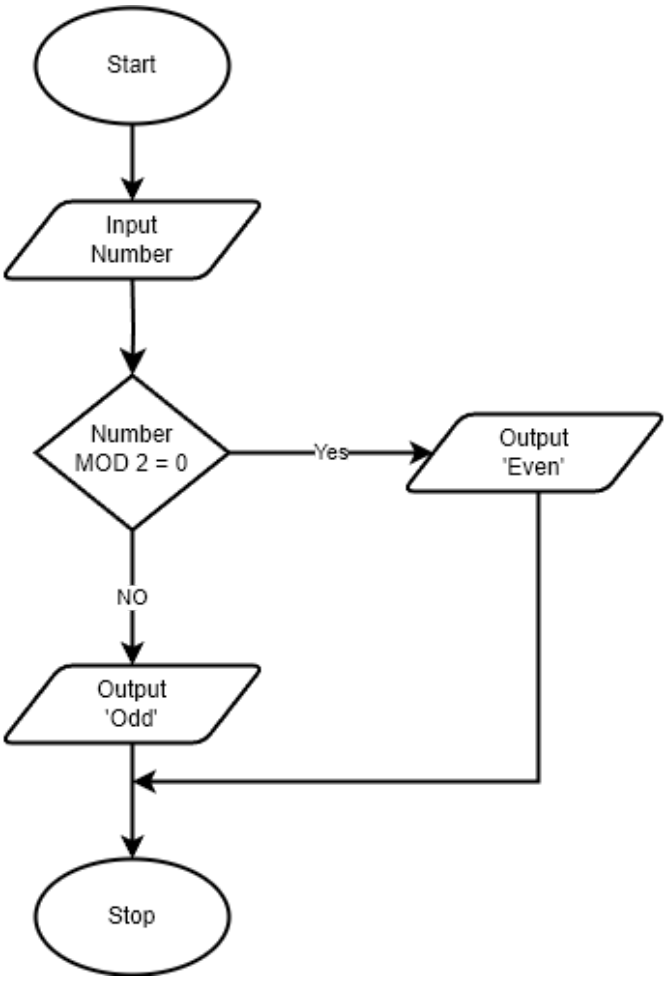
Step 2 – Flowchart structure

Pseudocode statement	Flowchart symbol	Purpose
INPUT Number	Parallelogram (Input/Output)	User enters a number
IF Number MOD 2 = 0	Diamond (Decision)	Checks if the number is divisible by 2
OUTPUT "Even"	Parallelogram (Input/Output)	Displays message if the condition is true
OUTPUT "Odd"	Parallelogram (Input/Output)	Displays message if the condition is false
Start/End	Oval	Denotes beginning and end of the process
Arrows	Lines with arrows	Indicate flow of control

Step 4 – Flowchart

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Stepwise refinement

What is stepwise refinement?

- Stepwise refinement is the process of **breaking down a complex problem** into smaller, more manageable sub-problems in a logical order
- Each sub-problem is **refined step by step** until it is simple enough to be solved with a single subroutine or module
- This method ensures that the overall problem can be solved by addressing each part individually, in a **structured and efficient** way

Relationship to decomposition and top-down design

- **Decomposition** is the general concept of breaking a problem down into smaller parts
- **Top-down design** is the strategy used to perform decomposition
- **Stepwise refinement** is the process used in top-down design to **gradually refine** each major task into simpler sub-tasks
- In other words, **stepwise refinement is how top-down design is implemented**

Benefits of stepwise refinement

- Helps developers **understand and organise** the structure of a program
- Makes **testing and debugging easier** through **unit testing** of individual subroutines
- Encourages **code reuse** by breaking tasks into reusable components
- Supports **collaborative development**, as tasks can be divided between team members
- Each subroutine should:
  - **Be clear and focused** on a single task
  - **Be simple** enough to implement directly
  - **Not need** further breakdown

Example: Calculating student grades

- Here's how stepwise refinement could be applied to a program designed to calculate student grades for a teacher's classes:

Top-level task:

- Calculate grades for all students in all classes

Stepwise refinement:

- **Step 1 – Calculate the grade for each assessment**

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- For each question:
  - Mark the question
  - Store the mark
- Sum the marks for all questions in the assessment

- **Step 2 – Calculate the average grade for each student**
  - Add together grades from all assessments
  - Divide by the number of assessments
  - Store the average
- **Step 3 – Repeat for each class**
  - For every student in the class:
    - Perform Steps 1 and 2
- This structured breakdown allows each task to be turned into a subroutine, simplifying the design, implementation, and testing of the overall program



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Data Types & Records

Contents

- Data types
- Record structures

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com



Your notes

Primitive data types

What are data types?

- A data type is a classification of data into groups according to the **kind of data they represent**
- Computers use different data types to represent different types of data in a program
- The basic data types include:
  - **Integer**: used to represent **whole numbers**, either positive or negative
    - Examples: 10, -5, 0
  - **Real**: used to represent **numbers with a fractional part**, either positive or negative
    - Examples: 3.14, -2.5, 0.0
  - **Char**: used to represent a **single character** such as a letter, digit or symbol
    - Examples: 'a', 'B', '5', '$'
  - **String**: used to represent a **sequence of alphanumerical characters**
    - Examples: "Hello World", "1234", "@#$%"
  - **Boolean**: used to represent **true or false** values
    - Examples: True, False
  - **Date**: used to store a **calendar date**
    - Example: DD/MM/YY
- It is important to choose the **correct data type for a given situation** to ensure accuracy and efficiency in the program
- In pseudocode and some other programming languages, **data types must be declared** before they can be used
  - The data types are declared with each **data item** to be used
  - Each data item is given a **unique** name called an **identifier**
  - DECLARE <identifier> : <data type>

Data Type	Example Value	Pseudocode	Python	Java	VB.NET

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Integer	10	DECLARE Age : INTEGER Age ← 10	age = 10	int age = 10;	Dim age As Integer = 10
Real	3.14	DECLARE Temp : REAL Temp ← 3.14	temp = 3.14	double temp = 3.14;	Dim temp As Double = 3.14
Char	'A'	DECLARE Grade : CHAR Grade ← 'A'	grade = 'A'	char grade = 'A';	Dim grade As Char = "A"c
String	"Hello"	DECLARE Name : STRING Name ← "Hello"	name = "Hello"	String name = "Hello";	Dim name As String = "Hello"
Boolean	TRUE	DECLARE LoggedIn : BOOLEAN LoggedIn ← TRUE	logged_in = True	boolean loggedIn = true;	Dim loggedIn As Boolean = True
Date	24/04/2025	DECLARE DOB : DATE DOB ← "24/04/2025"	dob = "24/04/2025" (string or datetime)	LocalDate dob = LocalDate.of(2025, 4, 24);	Dim dob As Date = #24/04/2025#



Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Purpose of records

What is a record?

- A record is a **composite data structure** used to store a **collection of related data items**
- Each item **can be** of a **different data type**
- It allows a programmer to group related values under **one identifier**
- Records help create a **more structured and readable** approach to storing and managing data
- A record contains a **fixed number of fields**, each with its own name and type
- In pseudocode, a record can by declared as:

```
TYPE <Typename>
   DECLARE <identifier> : <data type>
   DECLARE <identifier> : <data type>
   DECLARE <identifier> : <data type>
   ..
   ..
ENDTYPE
```

- Suppose we want to store information about a car
- We can define a **record type** like this:

```
TYPE Car
   DECLARE Make : STRING
   DECLARE Model : STRING
   DECLARE Colour : STRING
   DECLARE Price : REAL
   DECLARE DateOfRegistration : DATE
ENDTYPE
```

- We can then **create a variable** of this type:

```
DECLARE MyCar : Car
```

- And **assign values** to its fields:

```
MyCar.Make ← "Toyota"
MyCar.Model ← "Yaris"
MyCar.Colour ← "Red"
MyCar.Price ← 14995.99
MyCar.DateOfRegistration ← "12/03/2022"
```

- To **read from the record** we can:

```
OUTPUT "Make: ", MyCar.Make
OUTPUT "Model: ", MyCar.Model
OUTPUT "Colour: ", MyCar.Colour
OUTPUT "Price: £", MyCar.Price
OUTPUT "Date of Registration: ", MyCar.DateOfRegistration
```


© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com



Examiner Tips and Tricks

- **Records** in programming are a type of **data structure** used to group related data in your code
- These are **not the same** as records in a database, which refer to a **complete set of fields** on a single entity in a table (**row**)



Your notes

Key features of records

Feature	Explanation
Can store **different data types**	Unlike arrays, records are not limited to a single type
Fields are **named**	Each item in the record has a meaningful identifier
Offers **structured storage**	Easier to manage complex data about real-world entities

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Arrays

Contents

- Array basics
- Array pseudocode

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

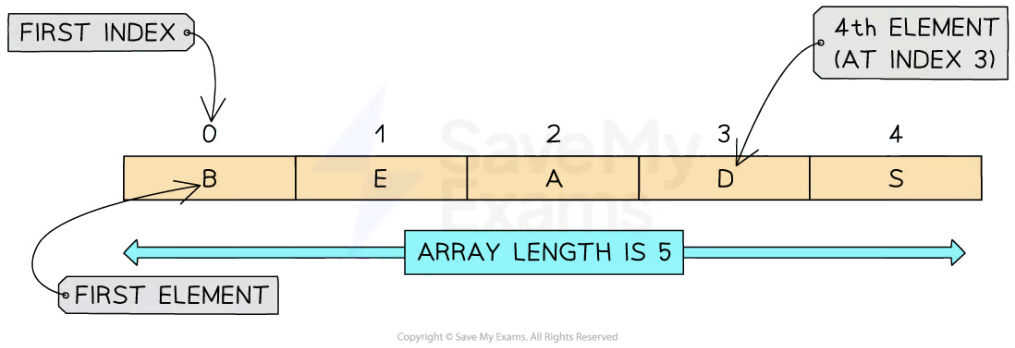
Arrays

What is an array?

- An array is an **ordered, static set of elements**
- Can only store **1 data type**
- The **position of each element** in an array is identified using the array's **index**
- The array's **first element** is the **lower bound (LB)**
- The array's **last element** is the **upper bound (UB)**
- The **lower bound of an array** is typically **0 or 1** depending on the language being used
- An array can be **one-dimensional** or **multi-dimensional**

One-dimensional (1D) arrays

- A 1D array is a **linear array**



Copyright © Save My Exams. All Rights Reserved

- To declare a 1D array in pseudocode you **must include** the lower bound, upper bound and data types:

```
DECLARE <identifier> : ARRAY[LB:UB] OF <data type>
```

- In this example a 1D array of **five elements** each containing **single character** can be declared as:

```
DECLARE Letters : ARRAY[0:4] OF CHAR
```

- An example **complete program** could be:

```
// Declare the array
DECLARE Letters : ARRAY[0:4] OF CHAR

// Assign values to each index
Letters[0] ← 'B'
Letters[1] ← 'E'
```


Study copy distributed by EduResource ZA | eduresourcza.com

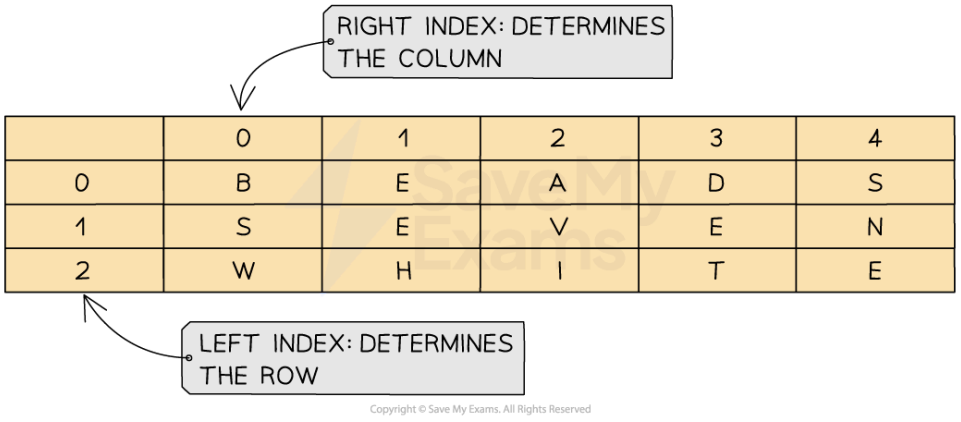
```
Letters[2] ← 'A'
Letters[3] ← 'D'
Letters[4] ← 'S'

// Output the full array
FOR Index ← 0 TO 4
   OUTPUT Letters[Index]
NEXT Index
```

- The array is declared with **indices** from 0 to 4
- Each element stores a single character using the CHAR **data type**
- The **loop** outputs each letter in order

Two-dimensional (2D) arrays

- A 2D array can be **visualised as a table**
- When navigating through a 2D array you first have to go **down the rows** and then **across the columns** to find a position within the array



- In 2D arrays the following must be declared:
  - Lower bound for rows (**LBR**) & upper bound for rows (**UBR**)
  - Lower bound for columns (**LBC**) & upper bound for columns (**UBC**)

```
DECLARE <identifier> : ARRAY[LBR:UBR, LBC:UBC] OF <data type>
```

- In this example a **2D array** can be declared as:

```
DECLARE Letters : ARRAY[0:2, 0:4] OF CHAR
```

- An example **complete program** could be:

```
DECLARE Letters : ARRAY[0:2, 0:4] OF CHAR

// Row 0
Grid[0,0] ← 'B'
Grid[0,1] ← 'E'
Grid[0,2] ← 'A'
Grid[0,3] ← 'D'
```

Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

```
Grid[0,4] ← 'S'

// Row 1
Grid[1,0] ← 'S'
Grid[1,1] ← 'E'
Grid[1,2] ← 'V'
Grid[1,3] ← 'E'
Grid[1,4] ← 'N'

// Row 2
Grid[2,0] ← 'W'
Grid[2,1] ← 'H'
Grid[2,2] ← 'I'
Grid[2,3] ← 'T'
Grid[2,4] ← 'E'
```

- ARRAY[0:2, 0:4] creates **3 rows** and **5 columns**
- First index is the **row**, second is the **column**: Letters[row, column]
- All elements are of type CHAR



Worked Example

A program reads data from a file and searches for specific data.

The main program needs to read 25 integer data items from the text file Data.txt into a local 1D array, DataArray

Write program code to declare the local array DataArray **[1]**

Answer

- 1D array with name DataArray (with 25 elements of type Integer) [1 mark]

Java

```
public static Integer[] DataArray = new Integer[25];
```

VB.NET

```
Dim DataArray(24) As Integer
```

Python

```
DataArray = [] #25 elements Integer
```



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com


Your notes

Searching arrays

What is a linear search?

- A linear search **starts with the first value** in a dataset and **checks every value one at a time** until all values have been checked
- A linear search can be performed **even if the values are not in order**

How do you perform a linear search?

Step	Instruction
1	Check the first value
2	**IF** it is the value you are looking for ▪ STOP!
3	**ELSE** move to the next value and check
4	**REPEAT UNTIL** you have checked all values and not found the value you are looking for

- A linear search can be used to **find elements in an array**
- Each element in the array is **checked in order**, from the **lower bound** to the **upper bound**, until the item is found or the upper bound is reached
- An example pseudocode algorithm to perform a linear search on a **1D array** could be:

```
DECLARE List : ARRAY[0:4] OF INTEGER
DECLARE Target : INTEGER
DECLARE Found : BOOLEAN
DECLARE Index : INTEGER

// Initialise array values
List[0] ← 12
List[1] ← 25
List[2] ← 37
List[3] ← 42
List[4] ← 56

// Input the value to search for
OUTPUT "Enter the value to search for:"
INPUT Target

Found ← FALSE
Index ← 0
```


© 2026 Save My Exams, Ltd. | Get more and ace your exams at savemyexams.com

Study copy distributed by EduResource ZA | eduresourcza.com

```
WHILE Index <= 4 AND Found = FALSE DO
   IF List[Index] = Target THEN
      Found ← TRUE
   ELSE
      Index ← Index + 1
   ENDIF
ENDWHILE

IF Found = TRUE THEN
   OUTPUT "Item found at index ", Index
ELSE
   OUTPUT "Item not found"
ENDIF
```

- Works on a fixed-size array List[0:4]
- Uses a WHILE loop for flexibility (early exit if found)
- Clearly separates **input**, **processing**, and **output**

Identifier table

Identifier	Data type	Description
List	ARRAY[0:4] OF INTEGER	Stores the list of numbers to search through
Target	INTEGER	The number the user wants to search for
Found	BOOLEAN	Tracks whether the target value has been found
Index	INTEGER	The current position being checked in the array

Sorting arrays

What is a bubble sort?

- A bubble sort is a simple sorting algorithm that starts at the beginning of a dataset and **checks values** in **'pairs'** and **swaps them** if they are **not in the correct order**
- One full run of comparisons from beginning to end is called a '**pass**', a bubble sort may require **multiple** '**passes**' to sort the dataset
- The algorithm is finished when there are **no more swaps to make**

How do you perform a bubble sort?

Step	Instruction
1	Compare the first two values in the dataset



Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

2	**IF** they are in the wrong order... ▪ Swap them
3	Compare the next two values
4	**REPEAT** step 2 & 3 until you reach the end of the dataset (**pass 1**)
5	**IF** you have made any swaps... ▪ **REPEAT** from the start (pass 2,3,4...)
6	**ELSE** you have not made any swaps... ▪ STOP! the list is in the correct order



Your notes

Example

- Perform a bubble sort on the following dataset

5	2	4	1	6	3

Step	Instruction
1	Compare the first two values in the dataset 5 | 2 | 4 | 1 | 6 | 3
2	**IF** they are in the wrong order... ▪ Swap them 2 | 5 | 4 | 1 | 6 | 3
3	Compare the next two values 2 | 5 | 4 | 1 | 6 | 3
4	**REPEAT** step 2 & 3 until you reach the end of the dataset ▪ 5 & 4 SWAP! 2 | 4 | 5 | 1 | 6 | 3

Study copy distributed by EduResource ZA | eduresourcza.com



Your notes

<table>
<tr><td></td><td>

5 & 1 SWAP!

<table>
<tr><td>2</td><td>4</td><td>1</td><td>5</td><td>6</td><td>3</td></tr>
</table>

5 & 6 NO SWAP!

<table>
<tr><td>2</td><td>4</td><td>1</td><td>5</td><td>6</td><td>3</td></tr>
</table>

6 & 3 SWAP!

<table>
<tr><td>2</td><td>4</td><td>1</td><td>5</td><td>3</td><td>6</td></tr>
</table>

End of pass 1

</td></tr>
<tr><td>5</td><td>
IF you have made any swaps...

REPEAT from the start
End of pass 2 (swaps made)

<table>
<tr><td>2</td><td>1</td><td>4</td><td>3</td><td>5</td><td>6</td></tr>
</table>

End of pass 3 (swaps made)

<table>
<tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr>
</table>

End of pass 4 (no swaps)

<table>
<tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr>
</table>
</td></tr>
<tr><td>6</td><td>
ELSE you have not made any swaps...

STOP! the list is in the correct order

</td></tr>
</table>

- A bubble sort can be used to **put elements in an array in order**
- A **pseudocode algorithm** for a bubble sort on a 1D array could be:

```
DECLARE Numbers : ARRAY[0:5] OF INTEGER
DECLARE Temp : INTEGER
DECLARE Pass : INTEGER
DECLARE Index : INTEGER
DECLARE Swapped : BOOLEAN

// Initialise the array
Numbers[0] ← 5
Numbers[1] ← 2
Numbers[2] ← 4
```


Study copy distributed by EduResource ZA | eduresourcza.com

```
Numbers[3] ← 1
Numbers[4] ← 6
Numbers[5] ← 3

// Bubble sort algorithm
FOR Pass ← 1 TO 5
  Swapped ← FALSE
  FOR Index ← 0 TO 4
    IF Numbers[Index] > Numbers[Index + 1] THEN
      Temp ← Numbers[Index]
      Numbers[Index] ← Numbers[Index + 1]
      Numbers[Index + 1] ← Temp
      Swapped ← TRUE
    ENDIF
  NEXT Index
  IF Swapped = FALSE THEN
    // List is already sorted
    EXIT
  ENDIF
NEXT Pass

// Output the sorted array
FOR Index ← 0 TO 5
  OUTPUT Numbers[Index]
NEXT Index
```

- Uses a fixed array of size 6
- Includes an **optimisation** to exit early if no swaps were made on a pass

Identifier table

Identifier	Data type	Description
Numbers	ARRAY[0:5] OF INTEGER	Stores the list of numbers to be sorted
Temp	INTEGER	Temporary variable used for swapping two elements in the array
Pass	INTEGER	Tracks the number of passes through the array
Index	INTEGER	Tracks the current position being compared in the array
Swapped	BOOLEAN	Indicates whether a swap occurred during a pass (used to optimise sorting)



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Files

Contents

- File handling

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Purpose of files

What is file handling?

- File handling is **the use of programming techniques** to work with information stored in **text files**
- Examples of file handing techniques are:
  - **opening** text files
  - **reading** text files
  - **writing** text files
  - **closing** text files

Concept	CIE A Level Pseudocode
Open file for reading	OPENFILE fruit.txt FOR READ
Open file for writing	OPENFILE Shopping.txt FOR WRITE
Close file	CLOSEFILE fruit.txt
Read line	READFILE fruit.txt, line
Write line	WRITE fruit.txt, "Oranges"
Check end of file	IF NOT EOF(fruit.txt) THEN
Create new file	OPENFILE Shopping.txt FOR WRITE (same as writing – it overwrites or creates)
Append to file	OPENFILE Shopping.txt FOR APPEND

- An example program written in pseudocode to:
  - **Opens** a text file for reading
  - **Reads** each line (a fruit name)
  - **Counts** how many fruits are in the file
  - **Outputs** the total

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- **Closes** the file

```
DECLARE FruitName : STRING
DECLARE FruitCount : INTEGER

FruitCount ← 0

OPENFILE FruitFile FOR READ

WHILE NOT EOF(FruitFile) DO
   READFILE FruitFile, FruitName
   FruitCount ← FruitCount + 1
ENDWHILE

CLOSEFILE FruitFile

OUTPUT "Total number of fruits: ", FruitCount
```

Identifier table

Identifier	Data type	Description
FruitName	STRING	Stores the name of each fruit read from the file
FruitCount	INTEGER	Keeps track of how many fruits have been read
FruitFile	FILE	The file containing the fruit names (e.g. "fruit.txt")

Worked Example

Write program code to read the contents of Data.txt into DataArray

Answer

- Opening file Data.txt to read [1 mark]
- Looping through all the 25/EOF ... [1 mark]
- ... reading each line and storing/appending into array [1 mark]
- Exception handling with appropriate output [1 mark]
- Closing the file (in an appropriate place) [1 mark]

Example program code:

Java

```
Integer Counter = 0;
try{
   Scanner Scanner1 = new Scanner(new File("Data.txt"));
   while(Scanner1.hasNextLine()){
      DataArray[Counter] = Integer.parseInt(Scanner1.next());
      Counter++;
   }
   Scanner1.close();
}catch(FileNotFoundException ex){
```

Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

```
    System.out.println("No data file found");
}
```

VB.NET

```
try
  Dim DataReader As New System.IO.StreamReader("Data.txt")
  Dim X As Integer = 0
  Do Until DataReader.EndOfStream
    DataArray(X) = DataReader.ReadLine()
    X = X + 1
  Loop
  DataReader.Close()
Catch ex As Exception
  Console.WriteLine("Invalid file")
End Try
```

Python

```
try:
  DataFile = open("Data.txt",'r')
  for Line in DataFile:
    DataArray.append(int(Line))
  DataFile.close()
except IOError:
  print("Could not find file")
```



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Introduction to Abstract Data Types (ADT)

Contents

- ADT overview

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

What is an abstract data type (ADT)?

- An abstract data type (ADT) is a **collection of data** and **a set of operations** on that data
- Three common ADT's are:
  - **Stacks**
  - **Queues**
  - **Linked lists**

Stack operations

What is a stack?

- A stack is an abstract data type that stores data using the **Last In, First Out (LIFO)** principle
- A stack is like **a pile of plates**, the last item you put on is the first one you take off
  - **PUSH**: Add an item to the top of the stack
  - **POP**: Remove the item from the top
- The **first item pushed** onto the stack will be the **last one popped** off
- A stack uses two pointers:
  - **A base pointer** - points to the first item in the stack
  - **A top pointer** - points to the last item in the stack

Main operations

Operation	Description
isEmpty()	This checks is the stack is empty by checking the value of the top pointer
push(value)	This adds a new value to the end of the list, it will need to check the stack is not full before pushing to the stack.
peek()	This returns the top value of the stack. First check the stack is not empty by looking at the value of the top pointer.
pop()	This removes and returns the top value of the stack. This first checks if the stack is not empty by looking at the value of the top pointer.

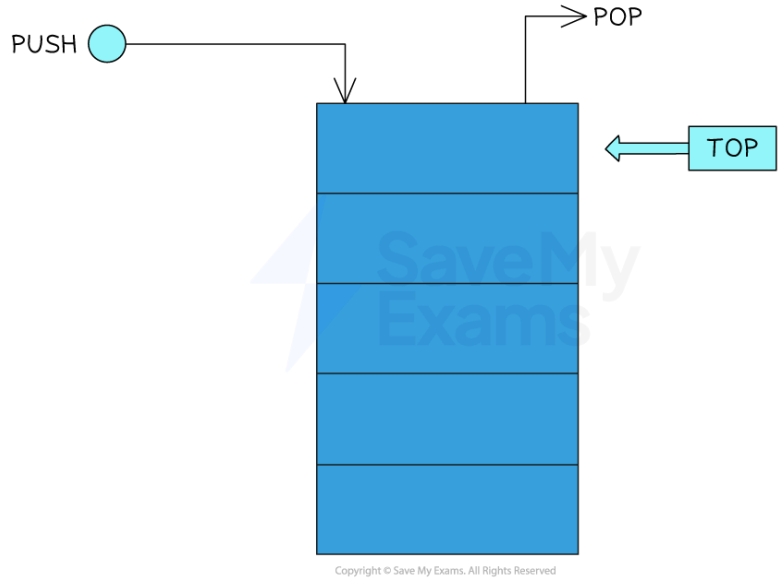
© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

size()	Returns the size of the stack
isFull()	Checks if the stack is full and returns the Boolean value, this compares the stack size to the top pointer.

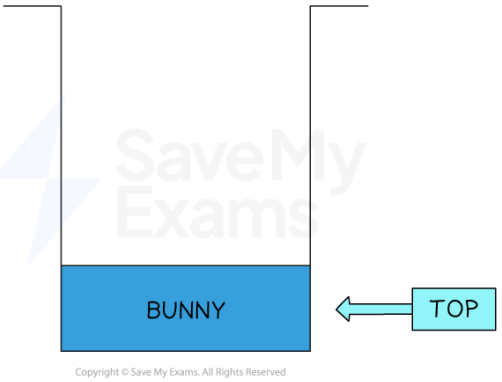
Your notes

- Stacks use a **pointer** which points to the top of the stack, where the next piece of data will be added (pushed) or the current piece of data can be removed (popped)



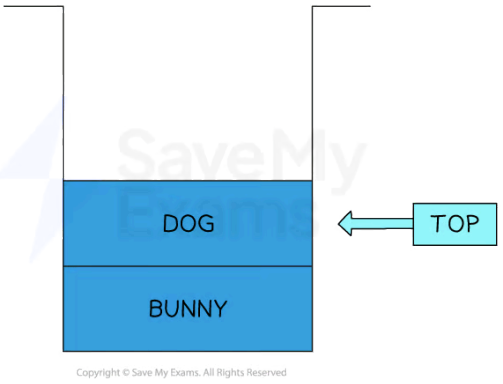
Pushing data to a stack

- When pushing (**adding**) data to a stack, the data is pushed to the position of the pointer
- Once pushed, the **pointer will increment by 1**, signifying the top of the stack
- Since the stack is a **static data structure**, an attempt to push an item on to a full stack is called a **stack overflow**
- This would push 'Bunny' onto the empty stack



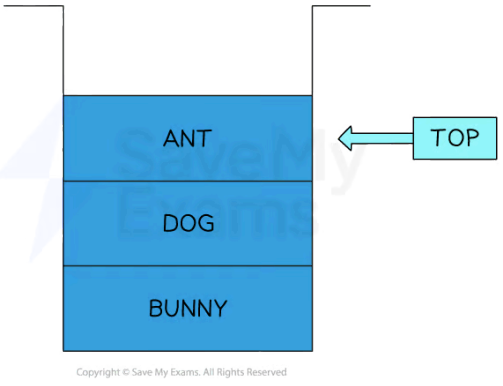
Study copy distributed by EduResource ZA | eduresourcza.com

- This would push 'Dog' to the top of the stack



Copyright © Save My Exams. All Rights Reserved

- This would push 'Ant' to the top of the stack



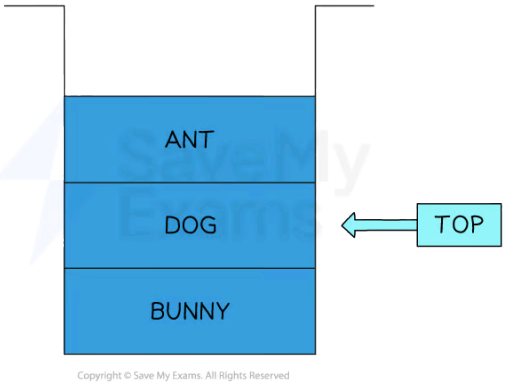
Copyright © Save My Exams. All Rights Reserved

Popping data from a stack

- When popping (**removing**) data from a stack, the data is popped from the position of the pointer
- Once popped, the **pointer will decrement by 1**, to point at the new top of the stack
- Since the stack is a static data structure, an attempt to pop an item from an empty stack is called a **stack underflow**
- This would pop 'Ant' from the top of the stack
- The new top of the stack would become 'Dog'

Your notes

Study copy distributed by EduResource ZA | eduresourcza.com



Copyright © Save My Exams. All Rights Reserved

Your notes

- Note that the data that is 'popped' isn't necessarily erased
- The pointer 'top' moves to show that 'Ant' is no longer in the stack and depending on the implementation, 'Ant' can be deleted or replaced with a null value, or left to be overwritten

Queue operations

What is a queue?

- A queue is an abstract data type that **stores data in the order it arrives**, using the **First In, First Out (FIFO)** principle
- A queue works like **a line of people waiting at a shop**, the first one in is the first one served
  - **ENQUEUE**: Add an item to the back of the queue
  - **DEQUEUE**: Remove the item from the front
- The **first item added** to the queue is the **first one removed**

Main operations

Operation	Description
enQueue(value)	Adding an element to the back of the queue
deQueue()	Returning an element from the front of the queue
peek()	Return the value of the item from the front of the queue without removing it from the queue
isEmpty()	Check whether the queue is empty

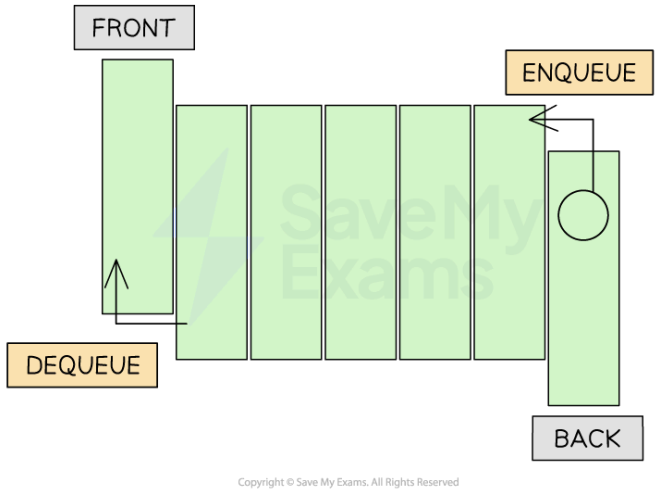
Study copy distributed by EduResource ZA | eduresourcza.com

isFull()	Check whether the queue is full (if the size has a constraint)

Your notes

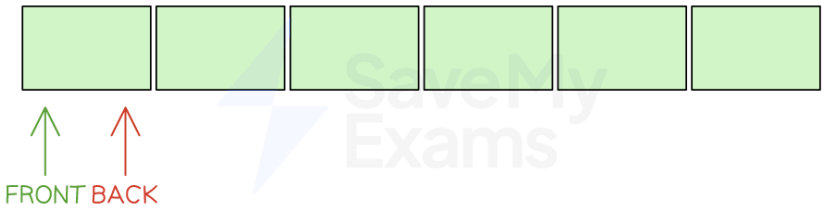
Linear queues

- A linear queue is a data structure that **consists of an array**
- Items are added to the **next available space** in the queue starting from the front
- Items are then **removed from the front** of the queue



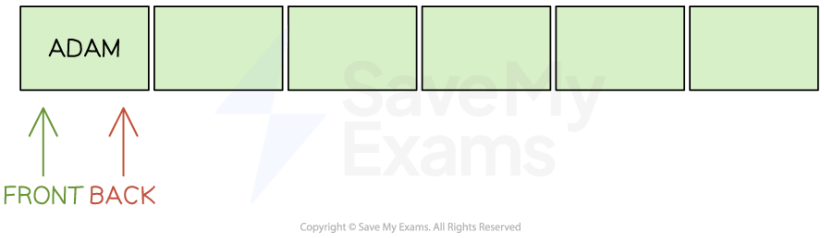
Copyright © Save My Exams. All Rights Reserved

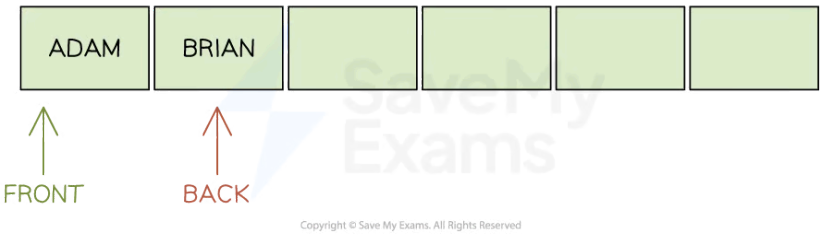
- Before **adding an item to the queue** you need to **check that the queue is not full**
- If the end of the array has not been reached, the rear **index pointer is incremented** and the new item is added to the queue
- In the example below, you can see the queue as the data Adam, Brian and Linda are enqueued.

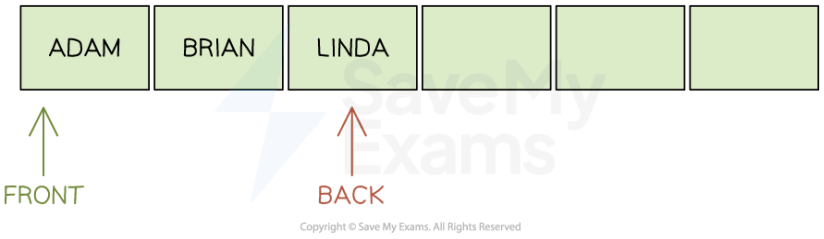


Copyright © Save My Exams. All Rights Reserved

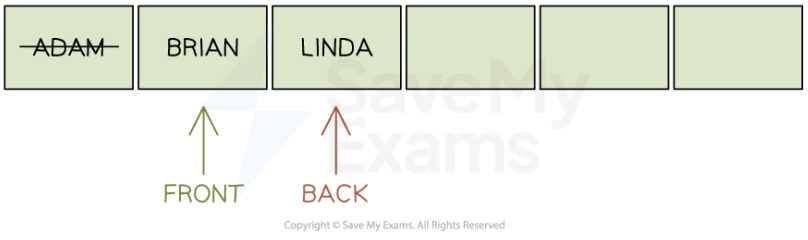
Study copy distributed by EduResource ZA | eduresourcza.com







- Before **removing** an item from the queue, you need to make sure that the queue is not empty
- If the queue is not empty the item at the front of the queue is returned and **the front is incremented by 1**
- In the example below, you can see the queue as the data Adam is dequeued

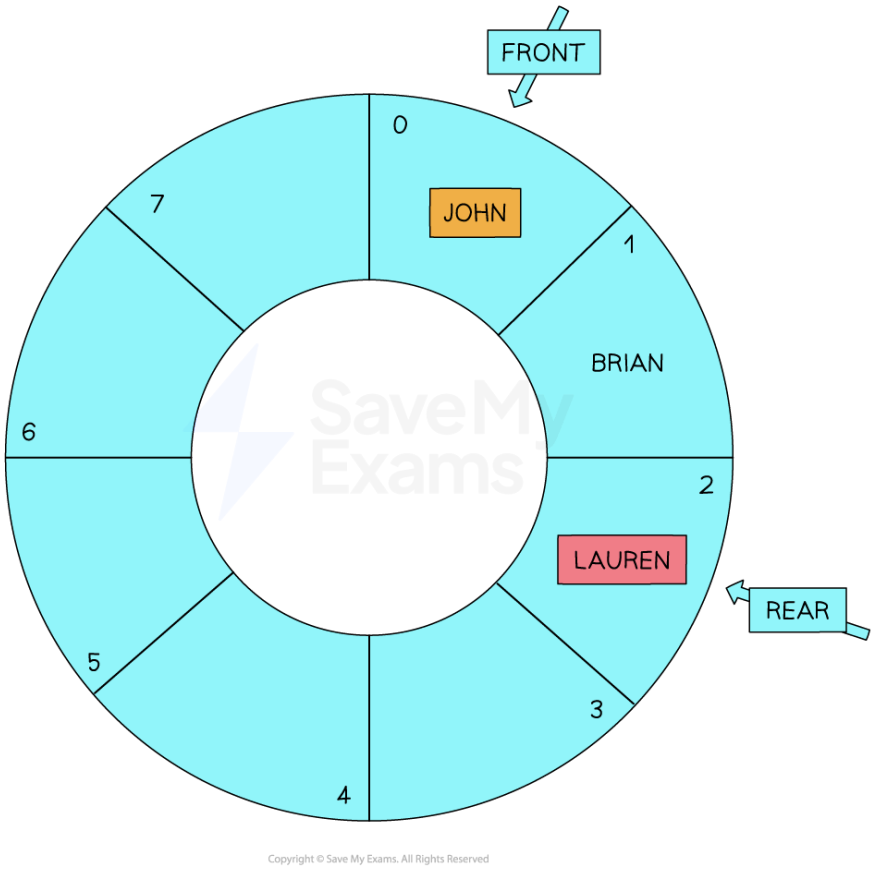


Circular queues

- A circular queue is **a static array** that has a **fixed capacity**

Study copy distributed by EduResource ZA | eduresourcza.com

- It means that as you add items to the queue you will **eventually reach the end** of the array
- When items are dequeued, **space is freed up at the start of the array**
- It would **take time** to move items up to the start of the array to free up space at the end, so a Circular queue (or circular buffer) is implemented
- It **reuses empty slots** at the front of the array that are caused when items are dequeued
- As items are enqueued and the rear index pointer reaches the last position of the array, it wraps around to point to the start of the array as long as it isn't full
- When items are dequeued the front index pointer will wrap around until it passes the rear index points which would show the queue is empty

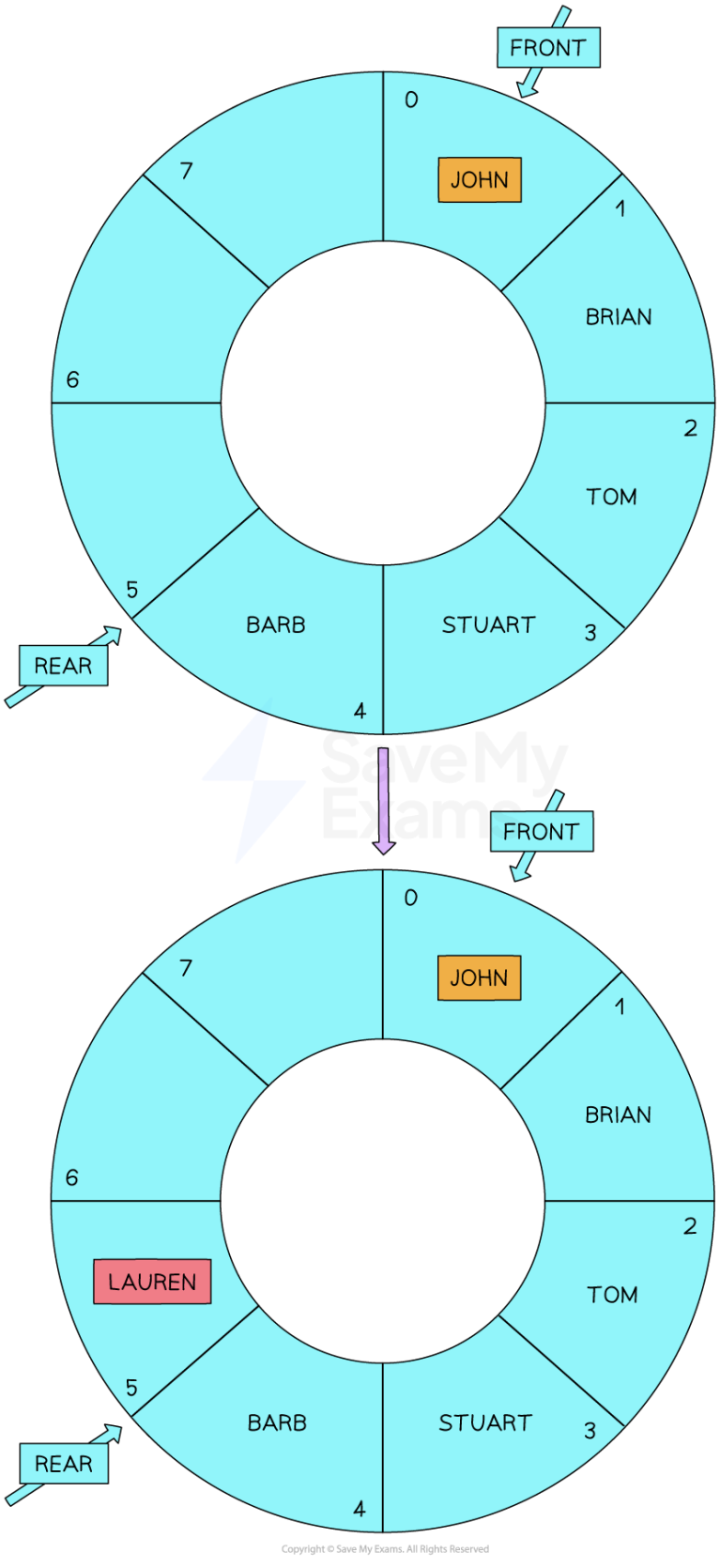


Copyright © Save My Exams. All Rights Reserved

- Before **adding** an item to the circular queue you need to check that the **array is not full**
- It will be full if the next position to be used **is already occupied** by the item at the front of the queue
- If the queue is not full then the **rear index pointer must be adjusted** to reference the next free position so that the new item can be added
- In the example below, you can see the circular queue as the data Lauren is enqueued.

Your notes

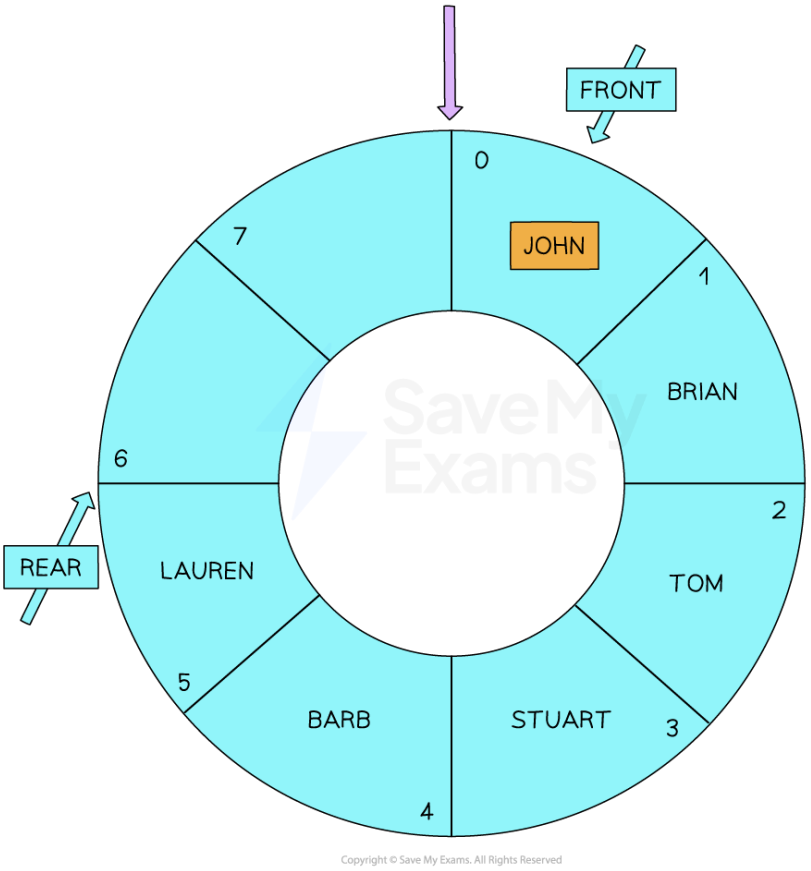
Study copy distributed by EduResource ZA | eduresourcza.com



Copyright © Save My Exams. All Rights Reserved

Your notes

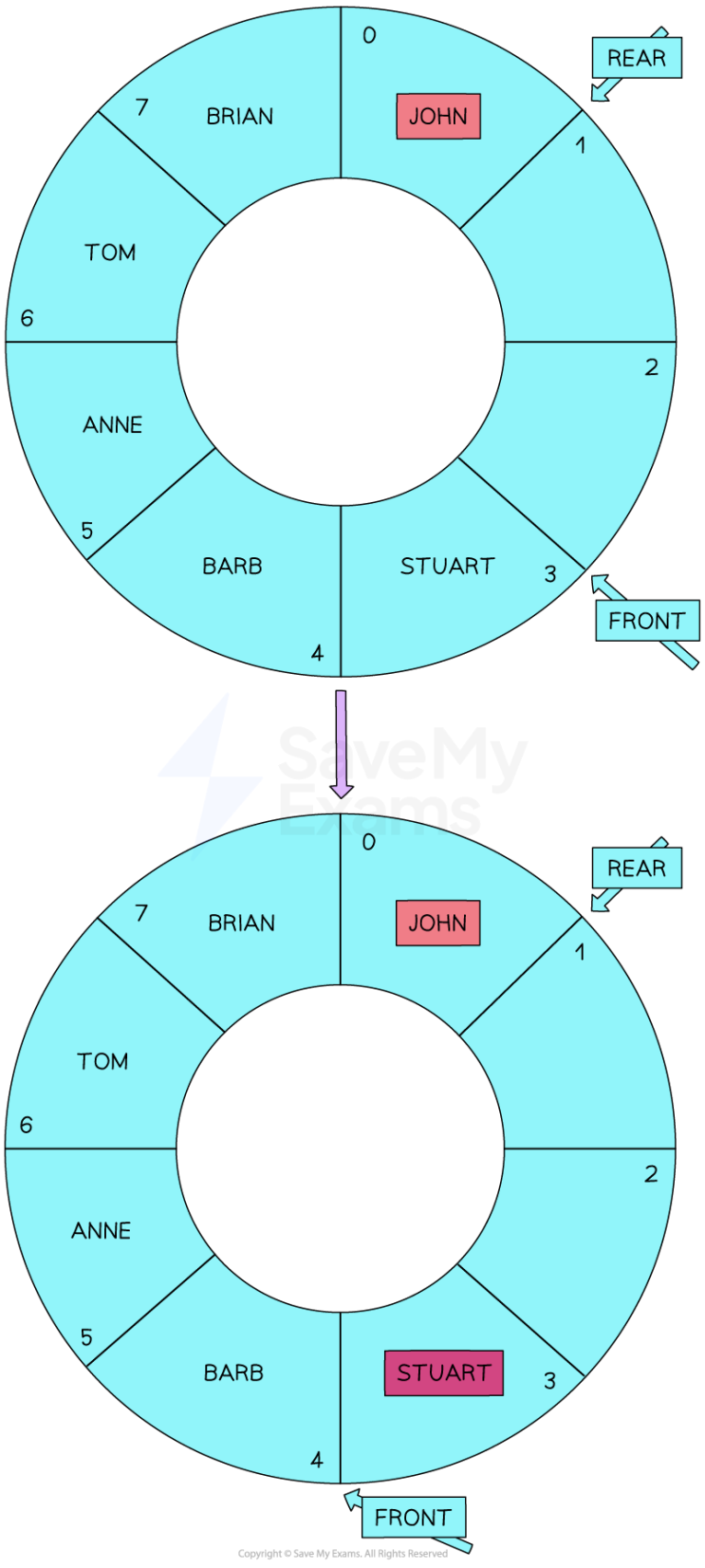
Study copy distributed by EduResource ZA | eduresourcza.com



Your notes

- Before **removing** an item from the queue, you need to make sure that the queue is not empty
- If the queue is not empty the **item at the front of the queue is returned**
- If this is the only item that was in the queue the **rear and front pointers are reset**, otherwise the pointer moves to reference the next item in the queue
- In the example below, you can see the queue as the data Adam is dequeued

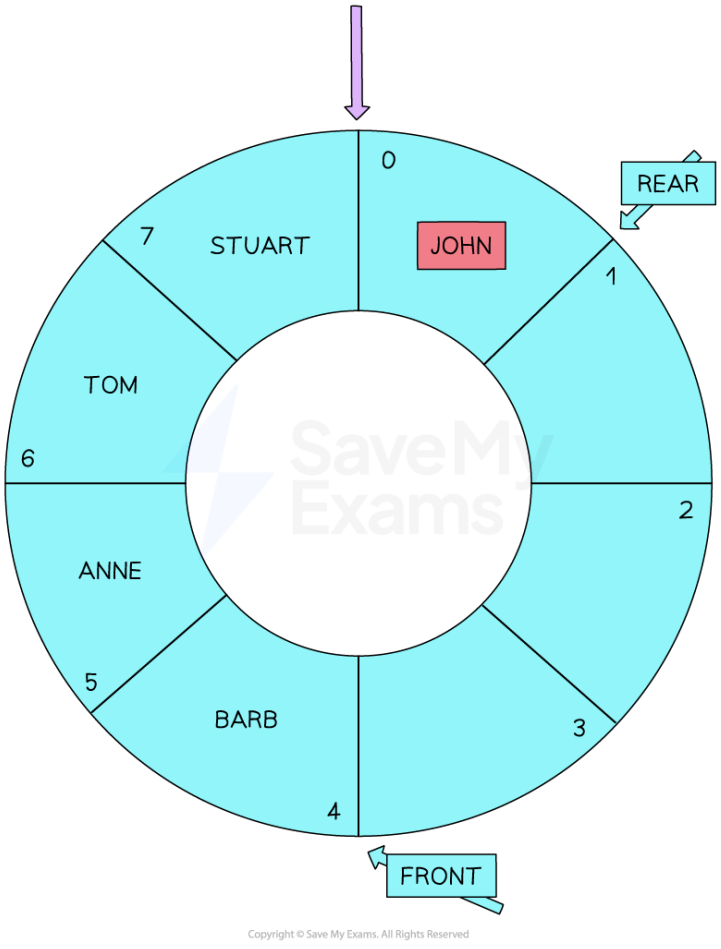
Study copy distributed by EduResource ZA | eduresourcza.com



Your notes

Copyright © Save My Exams. All Rights Reserved

Study copy distributed by EduResource ZA | eduresourcza.com



Copyright © Save My Exams. All Rights Reserved

Your notes

Linked list operations

What is a linked list?

- A linked list is an abstract data type where each item, or **node**, contains a **data field** and a **pointer to the next node** in the sequence
- A linked list is **a chain of items**, where each item stores **two things**:
  - The **actual data**
  - **A pointer** to the next item in the list
- New items are usually added to the **start** of the list, and each one **links** to the next
- Like a **scavenger hunt** where each clue tells you where to go next
- Each item is called a **node** and contains a data field alongside another address which is called a pointer
- For example:

Study copy distributed by EduResource ZA | eduresourcza.com

Index	Data	Pointer
0	'Apple'	2
1	'Pineapple'	0
2	'Melon'	-
3		

Start = 1 NextFree = 3

- The data field contains the **value of the actual data** which is part of the list
- The pointer field contains the **address of the next item** in the list

Examiner Tips and Tricks

- Linked lists can only be traversed by following each item's next pointer until the end of the list is located

Traverse a linked list

- Check if the linked list is empty
- Start at the node the pointer is pointing to (Start = 1)
- Output the item at the current node ('Pineapple')
- Follow the pointer to the next node repeating through each node until the end of the linked list.
- When the pointer field is empty/null it signals that the end of the linked list has been reached
- Traversing the example above would produce: 'Pineapple', 'Apple', 'Melon'

Adding a node

- An advantage of using linked lists is that values can easily be added or removed by editing the points
- The following example will add the word 'Orange' after the word 'Pineapple'

1. Check there is free memory to insert data into the node

2. Add the new value to the end of the linked list and update the 'NextFree' pointer

3	'Orange'	

Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Start = 1 NextFree = 4

3. The pointer field of the word 'Pineapple' is updated to point to 'Orange', at position 3

1	'Pineapple'	3

4. The pointer field of the word 'Orange' is updated to point to 'Apple' at position 0

3	'Orange'	0

4. When traversed this linked list will now output: 'Pineapple', 'Orange', 'Apple', 'Melon'

Removing a node

- Removing a node also involves updating nodes, this time to bypass the deleted node
- The following example will remove the word 'Apple' from the original linked list

1. Update the pointer field of 'Pineapple' to point to 'Melon' at index 2

Index	Data	Pointer
0	'Apple'	2
1	'Pineapple'	2
2	'Melon'	-

2. When traversed this linked list will now output: 'Pineapple', 'Melon'

- The node is not truly removed from the list; it is only ignored
- Although this is easier it does waste memory
- Storing pointers themselves also means that more memory is required compared to an array
- Items in a linked list are also stored in a sequence, they can only be traversed within that order so an item cannot be directly accessed as is possible in an array



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Programming Basics

Contents

- Pseudocode basics

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Variables & Constants

What is a variable?

- A variable is an **identifier** that **can change** in the lifetime of a program
- Identifiers should be:
  - In **mixed case** (Pascal case)
  - Only **contain letters** (A-Z, a-z)
  - Only **contain digits** (0–9)
  - **Start with a capital** letter and not a digit
- A variable can be **associated a datatype** when it is declared
- When a variable is declared, **memory is allocated** based on the data type indicated

Pseudocode
DECLARE <identifier> : <datatype>

- To declare a variable, use the DECLARE keyword followed by the **name** and **data type**:

```
DECLARE Age : INTEGER
DECLARE Name : STRING
DECLARE IsLoggedIn : BOOLEAN
DECLARE Temperature : REAL
DECLARE DOB : DATE
```

- You can then **assign a value** using the assignment operator ← :

```
Age ← 18
Name ← "Alice"
IsLoggedIn ← TRUE
```

What is a constant?

- A constant is an **identifier set once in the lifetime of a program**
- Constants are generally named in **all uppercase characters**
- Constants **aid** the **readability** and **maintainability**

Pseudocode
CONSTANT <identifier> ← <value>

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- To declare a constant, use the CONSTANT keyword:

```
CONSTANT Pi ← 3.14159
CONSTANT MaxScore ← 100
CONSTANT SchoolName ← "Meridian Academy"
```

- Constants are **not reassigned** during execution – they are fixed values used throughout the algorithm



Your notes

Example

- You are writing a program to calculate the area of a circle using the formula:

$Area \leftarrow \pi \times radius^2$

- π is a **constant** (it never changes)
- radius and area are **variables** (they change depending on the input)

```
CONSTANT Pi ← 3.14159

DECLARE Radius : REAL
DECLARE Area : REAL

OUTPUT "Enter the radius of the circle:"
INPUT Radius

Area ← Pi * Radius * Radius

OUTPUT "The area of the circle is: ", Area
```

Variables and constants in different languages

Feature	Python	VB.NET	Java
Declare variable	Just assign it (no keyword needed)	Dim radius As Double	double radius;
Assign variable	radius = 5	radius = 5	radius = 5;
Declare constant	Convention: use UPPERCASE (PI = 3.14) (not enforced)	Const Pi As Double = 3.14159	final double PI = 3.14159;
Reassign constant?	Yes (not truly constant unless enforced)	Cannot change once set	Cannot change once set
Use in calc.	area = PI * radius * radius	area = Pi * radius * radius	area = PI * radius * radius;

Arithmetic & logical operators

Arithmetic operators

Study copy distributed by EduResource ZA | eduresourcza.com

- Arithmetic operators are used to perform basic **maths operations** in a program
- These include adding, subtracting, multiplying and dividing values



Your notes

Common arithmetic operators

Operator	Purpose	Example	Result
+	Addition or string concatenation	5 + 3	8
		"John" + " " + "Doe"	"John Doe"
-	Subtraction	10 - 4	6
*	Multiplication	3 * 5	15
/	Division	10 / 2	5
^	Exponentiation (power of)	3 ^ 3	27
MOD	Modulus (remainder after division)	10 MOD 3	1

Operator precedence (BODMAS / BIDMAS)

- Arithmetic operators follow **operator precedence**
- Multiplication and division happen before addition and subtraction unless brackets are used

```
result ← 2 + 3 * 4        // gives 14
result ← (2 + 3) * 4      // gives 20
```

Logical operators

- Logical operators are used to **compare values**
- They return either **TRUE** or **FALSE** and are commonly used in conditions and loops

Common logical operators

Operator	Purpose	Example	Result
=	Equal to	5 = 6	FALSE
<>	Not equal to	5 != 7	TRUE
>	Greater than	5 > 10	FALSE

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

<	Less than	5 < 10	TRUE
>=	Greater than or equal to	5 >= 10	FALSE
<=	Less than or equal to	5 <= 10	TRUE



Your notes

Example code

```
x ← 5
y ← 10

OUTPUT x == y       // FALSE
OUTPUT x <> y       // TRUE
OUTPUT x < y        // TRUE
OUTPUT x > y        // FALSE
OUTPUT x <= y       // TRUE
OUTPUT x >= y       // FALSE
```


© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Constructs

Contents

- Selection
- Iteration

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Selection

What is selection?

- Selection is when the **flow of a program is changed,** depending on a **set of conditions**
- The outcome of this condition will then determine **which lines or block of code is run next**
- Selection is used for **validation**, **calculation** and making sense of a **user's choices**
- There are two ways to write selection statements:
  - **if**... **then**... **else**...
  - **case**...

If statements

What is an if statement?

- An If statements allow you to **execute a set of instructions if a condition is true**
- They have the following syntax:

Pseudocode	
IF <condition> THEN <statement(s)> ENDIF	IF <condition> THEN <statement(s)> ELSE <statement(s)> ENDIF
Without an ELSE clause	With an ELSE clause

Nested if statements

- Nested if statements are **an if statement within an if statement**
- Nested means to be '**stored inside the other**'

Example code

```
IF Player2Score > Player1Score THEN
   IF Player2Score > HighScore THEN
      OUTPUT Player2, " is champion and highest scorer"
   ELSE
      OUTPUT Player2, " is the new champion"
   ENDIF
ELSE
```


© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

```
    OUTPUT Player1, " is still the champion"
    IF Player1Score > HighScore THEN
        OUTPUT Player1, " is also the highest scorer"
    ENDIF
ENDIF
```



Your notes

If statements in different languages

Python	VB.net

weather = input("What is the weather like today? (sunny, rainy, snowy): ")
 	
if weather == "sunny":
  print("Don't forget your sunglasses!")
elif weather == "rainy":
  print("Take an umbrella with you.")
elif weather == "snowy":
  print("Wear warm clothes!")
else:
  print("Not sure what to suggest for that kind of weather.")
	Module WeatherCheck
	  Sub Main()
	  Dim weather As String
	
	  Console.Write("What is th
	  weather = Console.ReadL
	
	  If weather = "sunny" Ther
	  Console.WriteLine("Dor
	  ElseIf weather = "rainy" Th
	  Console.WriteLine("Tak
	  ElseIf weather = "snowy"
	  Console.WriteLine("We
	  Else
	  Console.WriteLine("Not
	  End If
	
	  Console.ReadLine() ' Keep
	  End Sub
	End Module

Case statements

What is a case statement?

- A case statement **can mean less code** but it only useful when comparing **multiple values of the same variable**
- **If** statements are **more flexible** and are generally used more in languages such as Python
- The format of a CASE statement is:

Pseudocode	
CASE of <identifier> <value 1> : <statement1> <statement2> ... <value 2> : <statement1> <statement2> ...	CASE of <identifier> <value 1> : <statement1> <statement2> ... <value 2> : <statement1> <statement2> ...

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

... ENDCASE	OTHERWISE : <statement1> <statement2> ... ENDCASE
	An OTHERWISE clause can be the last clause



Your notes

Example code

```
DECLARE Direction : STRING

OUTPUT "Enter a direction (N, S, E, W):"
INPUT Direction

CASE OF Direction
   "N" : OUTPUT "You are heading North"
   "S" : OUTPUT "You are heading South"
   "E" : OUTPUT "You are heading East"
   "W" : OUTPUT "You are heading West"
   OTHERWISE : OUTPUT "Invalid direction entered"
ENDCASE
```

Case statements in different languages

Python	VB.net
weather = input("What is the weather like today? (sunny, rainy, snowy): ") match weather: case "sunny": print("Don't forget your sunglasses!") case "rainy": print("Take an umbrella with you.") case "snowy": print("Wear warm clothes!") case _: print("Not sure what to suggest for that kind of weather.")	Module WeatherCheck Sub Main() Dim weather As String Console.Write("What is th weather = Console.ReadL Select Case weather Case "sunny" Console.WriteLine("D Case "rainy" Console.WriteLine("T Case "snowy" Console.WriteLine("W Case Else Console.WriteLine("N End Select Console.ReadLine() ' Keep End Sub End Module

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

What is iteration?

- Iteration is **repeating a line or a block of code** using a loop
- Iteration can be:
  - **Count-controlled**
  - **Condition-controlled**

Count-controlled loops

What is a count-controlled loop?

- A count-controlled loop is when the code is repeated **a fixed number of times** (e.g. using a FOR loop)
- A count-controlled loop can be written as:

Pseudocode	
FOR <identifier> ← <value1> TO <value2> <statements(s)> NEXT <identifier>	FOR <identifier> ← <value1> TO <value2> STEP <increment> <statements(s)> NEXT <identifier>

- In a **FOR loop**, the **increment** must be an **expression that evaluates to an integer**
  - The loop **starts** at value1
  - The identifier is updated by the **increment** value on each iteration
  - The loop continues until the identifier **passes value2**
- The increment can be **positive** or **negative**, depending on the direction you want the loop to count

Nested count-controlled loops

```
FOR i ← 1 TO 3
  FOR j ← 1 TO 3
    OUTPUT i, " x ", j, " = ", i * j
  NEXT j
NEXT i
```

- The **outer loop** runs i from 1 to 3
- For **each value of** i, the **inner loop** runs j from 1 to 3
- It outputs all combinations of i × j

```
1 x 1 = 1
1 x 2 = 2
```


© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

1 x 3 = 3
2 x 1 = 2
...
3 x 3 = 9



Your notes

Count-controlled loops in different languages

Python	VB.net	Java
Print the numbers 1 to 5		

for i in range(1, 6):
  print(i)
	For i As Integer = 1 To 5
	  Console.WriteLine(i)
	Next
		for (int i = 1; i <= 5; i++) {
		  System.out.println(i);
		}

Condition-controlled loops

What is a condition controlled loop?

- A condition controlled loop is when the code is repeated **until a condition is met**
- There are two types of condition controlled loops:
  - **Post-condition (REPEAT)**
  - **Pre-condition (WHILE)**

Post-condition loops (REPEAT)

- A post-condition loop **is executed at least once**
- The condition must be an **expression** that evaluates to a **Boolean** (True/False)
- The condition is tested **after the statements are executed** and only stops once the condition is evaluated to True
- It can be written as:

Pseudocode
REPEAT   <statement(s)> UNTIL <condition>

Pre-condition loops (WHILE)

- The condition must be an **expression** that evaluates to a **Boolean** (True/False)
- The condition is tested and statements are **only executed if the condition evaluates to True**
- After statements have been executed the condition is **tested again**

Study copy distributed by EduResource ZA | eduresourcza.com

- The loop ends when the condition **evaluates to False**
- It can be written as:

Pseudocode
WHILE <condition> DO   <statement(s)> ENDWHILE



Your notes

Nested condition-controlled loops

```
DECLARE Username : STRING
DECLARE Count : INTEGER

Count ← 0

WHILE Count < 3
   Username ← ""

   WHILE Username = ""
      OUTPUT "Enter a username: "
      INPUT Username

      IF Username = "" THEN
         OUTPUT "Username cannot be blank."
      ENDIF
   ENDWHILE

   OUTPUT "Username accepted: ", Username
   Count ← Count + 1
ENDWHILE
```

- Ask the user to **enter up to 3 usernames**
- For each one, keep asking **until a non-empty name** is entered
- **Outer WHILE** runs 3 times (for 3 usernames)
- **Inner WHILE** ensures that each username is **not blank**

Condition-controlled loops in different languages

Python	VB.net	Java
Keep asking the user to enter a password until they type "ac		

Study copy distributed by EduResource ZA | eduresourcza.com

```
password = ""

while password != "admin":
    password = input("Enter the password: ")

    if password != "admin":
        print("Incorrect, try again.")

print("Access granted!")
```

```
Module PasswordCheck
    Sub Main()
        Dim password As String = ""

        While password <> "admin"
            Console.Write("Enter the password: ")
            password = Console.ReadLine()

            If password <> "admin" Then
                Console.WriteLine("Incorrect, try again.")
            End If
        End While

        Console.WriteLine("Access granted!")
        Console.ReadLine() ' Keeps the console open
    End Sub
End Module
```

```
import

public c
    publi
        Sca
        Str

        wh
            S
            p

            i

            }
        }

        Sys
        sca
    }
}
```

Your notes

When to use each type of loop

Loop type	When to use	Example scenario
Count-controlled loop (FOR)	When you know **in advance** how many times you want the loop to run	Repeating an action 5 times, generating a multiplication table
Pre-condition loop (WHILE)	When you want to **check a condition before** running the loop. It may run **0 or more** times	Keep asking for a valid password before granting access
Post-condition loop (REPEAT ... UNTIL)	When you want the loop to run **at least once**, and then stop when a condition becomes true	Ask the user for input and validate it after first run

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Structured Programming

Contents

- Procedures & functions

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

What are functions and procedures?

- Functions and procedures are a type of **sub program**, a **sequence of instructions** that **perform a specific task** or **set of tasks**
- Procedures and functions are **defined at the start of the code**
- Sub programs are often used to **simplify** a program by **breaking it into smaller, more manageable parts**
- Sub programs can be used to:
  - **Avoid duplicating code** and can be reused throughout a program
  - **Improve** the **readability** and **maintainability** of code
  - Perform **calculations**, to **retrieve data**, or to **make decisions** based on input
- **Parameters** are **values** that are **passed into a sub program**
  - Parameters can be variables or values and they are located **in brackets** after the name of the sub program
  - Example: FUNCTION TaxCalculator(pay,taxcode) **OR** PROCEDURE TaxCalculator(pay,taxcode)
- Sub programs **can have multiple parameters**
- To use a sub program you '**call**' it from the main program

What's the difference between a function and procedure?

- A **Function returns a value** whereas a procedure does not

Procedures

- Procedures are defined using the **PROCEDURE** keyword in pseudocode
- A procedure can be written as:

Pseudocode	
PROCEDURE <identifier>() <statement(s)> ENDPROCEDURE	PROCEDURE <identifier>(<param1>:<data type>, <param2>:<data type>...) <statement(s)> ENDPROCEDURE

- The identifier is used to **call the procedure**
- To call a procedure, it can be written as:

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

CALL <identifier>()
CALL <identifier>(Value1,Value2,...)

- When parameters are used, value1, value2... must be of the correct data type and in the same sequence

Example code

```
PROCEDURE GreetUser(Name : STRING)
   OUTPUT "Hello, ", Name, "!"
ENDPROCEDURE

DECLARE UserName : STRING

OUTPUT "Enter your name: "
INPUT UserName

CALL GreetUser(UserName)
```

- PROCEDURE GreetUser defines the procedure, which takes one parameter: Name
- The procedure prints a personalised greeting
- The main program asks for the user's name and then **calls** the procedure using CALL

Functions

- Functions are defined using the **FUNCTION** keyword in pseudocode or **def** keyword in Python
- Functions **always return a value** so they **do not** use the CALL function, instead they are called **within an expression**
- A function can be written as:

	Pseudocode
FUNCTION <identifier> RETURNS <data type> <statement(s)> ENDFUNCTION	FUNCTION <identifier>(<param1>:<data type>, <param2> <statement(s)> ENDFUNCTION

Example code

```
FUNCTION calculate_area(length: INTEGER, width: INTEGER)
  area ← length * width
  RETURN area
ENDFUNCTION

# Output the value returned from the function
OUTPUT(calculate_area(5,3))
```

- Defines a **function** called calculate_area
- It takes **two parameters**: length and width, both of type INTEGER
- A local variable area is declared and assigned the result of length × width

Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

- The function **returns** the result to wherever it was called from



Your notes

Parameter passing

What is parameter passing?

- Parameter passing is the method by which **values or references** are given to procedures or functions so they can **use or modify data during execution**
- When writing **procedures**, you can control how parameters are passed using:
  - BYVAL – pass **a copy** of the value (default behaviour)
  - BYREF – pass **the actual variable**, so changes inside the procedure affect the original

Keyword	What it does
BYVAL	Passes a **copy** of the value (no change to original)
BYREF	Passes a **reference** to the variable (can be modified)

- If **no keyword** is used, it is assumed to be BYVAL
- **Functions** should **not use** BYREF **parameters** – only **procedures** should

Example – Swapping two values (Procedure with BYREF)

```
PROCEDURE Swap(BYREF X : INTEGER, Y : INTEGER)
   DECLARE Temp : INTEGER
   Temp ← X
   X ← Y
   Y ← Temp
ENDPROCEDURE
```

- X is passed **by reference**, so it **can be changed**
- Y is passed **by value**, so any change to Y inside the procedure **does not affect** the original value



Worked Example

A video-conferencing program supports up to six users. Speech from each user is sampled and digitised (converted from analogue to digital). Digitised values are stored in array Sample.

The array Sample consists of 6 rows by 128 columns and is of type integer. Each row contains 128 digitised sound samples from one user.

The digitised sound samples from each user are to be processed to produce a single value which will be stored in a 1D array Result of type integer. This process will be implemented by procedure Mix().

© 2026 Save My Exams, Ltd.

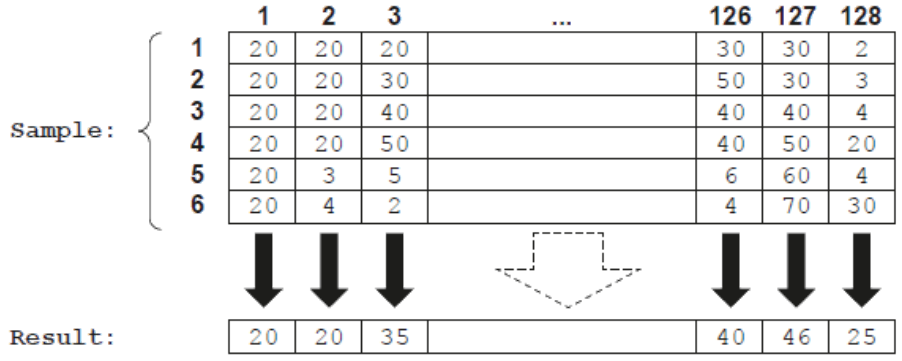
Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

A procedure Mix() will:

- calculate the average of each of the 6 sound samples in a column
- ignore sound sample values of 10 or less
- store the average value in the corresponding position in Result
- repeat for each column in array Sample

The diagram uses example values to illustrate the process:



Write pseudocode for procedure Mix().

Assume Sample and Result are global. **[6]**

Answer

```
PROCEDURE Mix()
 DECLARE Count, Total ThisNum : INTEGER
 DECLARE ThisUser, ThisSample : INTEGER

 FOR ThisSample ← 1 TO 128
  Count ← 0
  Total ← 0
  FOR ThisUser ← 1 TO 6
   IF Sample[ThisUser, ThisSample] > 10 THEN
    Count ← Count + 1
    Total ← Total + Sample[ThisUser, ThisSample]
   ENDIF
  NEXT ThisUser
  Result[ThisSample] ← INT(Total / Count)
 NEXT ThisSample

ENDPROCEDURE
```

- Declaration **and** initialisation before inner loop of Count **and** Total [1 mark]
- Outer Loop for 128 iterations [1 mark]
- Inner loop for six iterations [1 mark]
- Test for sample > 10 **in a loop** [1 mark]
- and if true sum Total and increment Count [1 mark]
- Calculate average value and assign to Result array **after inner loop and within outer loop** [1 mark]
- Use of INT()/ DIV to convert average to integer [1 mark]

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Program Development Life cycle

Contents

- Development models

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

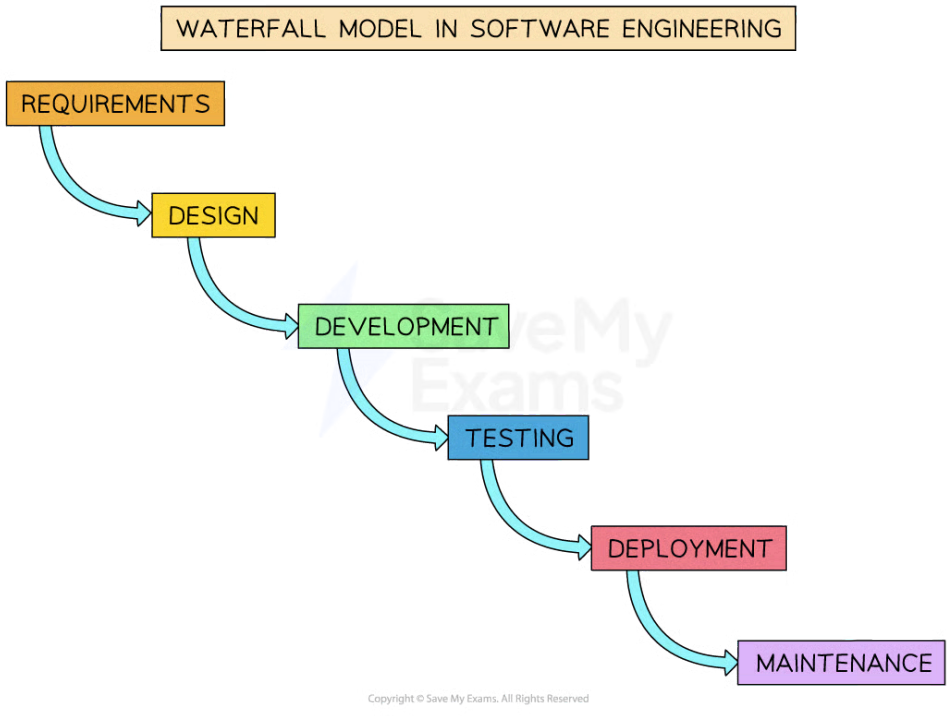
Waterfall

What is the waterfall model?

- The Waterfall Model is a **sequential software development process** divided into **distinct phases**
- Each phase **must be completed** before the next one begins

Steps

1. **Requirement Gathering and Analysis:** All possible system requirements to be developed are captured and documented clearly
2. **System Design:** The requirements are translated into a design. Architects and designers define the overall architecture and identify the main components
3. **Implementation:** The actual code is written in this phase based on the design documents, turning the system design into a functional program
4. **Integration and Testing:** All the components and modules are integrated and tested to ensure that the entire system works as expected
5. **Deployment:** The product is released to the market or handed over to the client. It may involve installation, customization, and training
6. **Maintenance:** Post-release, the system needs regular maintenance to fix bugs, improve performance, or add new features



Study copy distributed by EduResource ZA | eduresourcza.com



Your notes

Benefits and drawbacks

Benefits	Drawbacks
Simple and linear – easy to understand and follow	**Inflexible** – difficult to make changes once development begins
Clear stages and milestones – easy to track progress	**Expensive to fix late problems** – issues found late are harder to resolve
Ideal for well-defined projects – works best when requirements are fixed	**Long development cycle** – each stage must be completed before moving on

Suitability

- The Waterfall Model is most suitable for projects where **requirements are well understood** and **unlikely to change**
- It works well when **high quality and compliance are essential**, and there is a clear understanding of the project's goals and constraints

Iterative (Agile)

What is the iterative model?

- The iterative model s a **type of Agile software development methodology** that promotes **adaptability** and high customer involvement

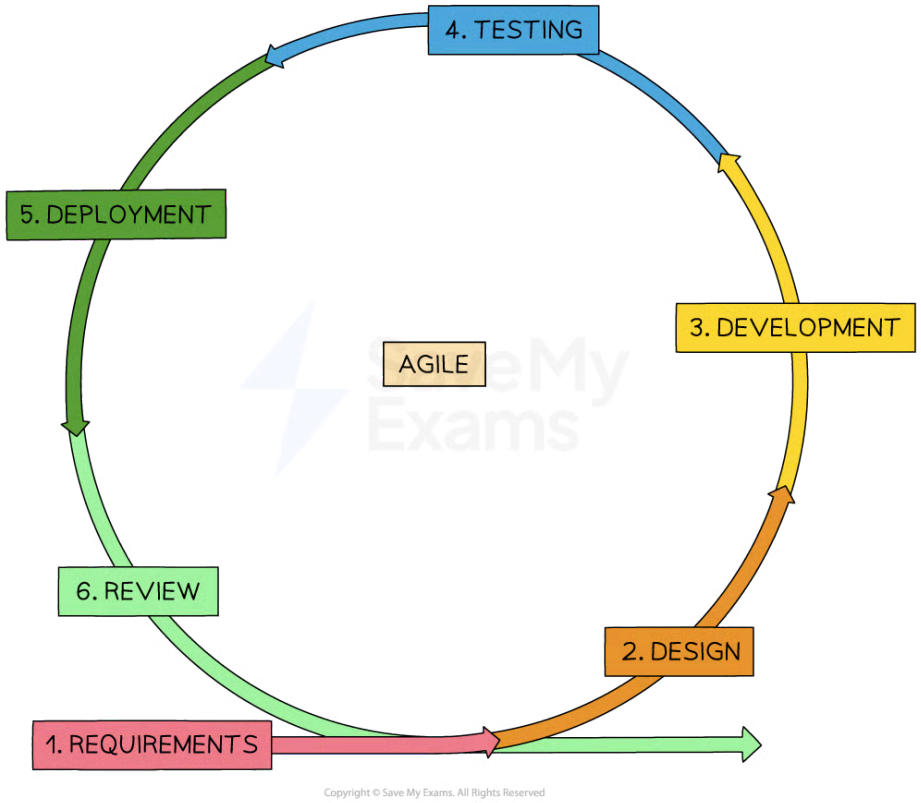
Steps

1. **Identify user stories and requirements**
    1. Work closely with stakeholders to gather functional and non-functional requirements
    2. Requirements are often written as **user stories** (e.g. *As a user, I want to...*)
2. **Plan the sprint (Sprint Planning)**
    1. Break down requirements into **tasks**
    2. Choose a set of tasks (features) for the **current sprint** (a short time-boxed development period, usually 1-4 weeks)
    3. Define the **sprint goal**
3. **Design the solution**
    1. Decide how the selected features will be built
    2. Focus is on **simple and adaptable design**, not heavy upfront documentation

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

4. **Develop the features**
    1. Write code for the selected tasks in the sprint
    2. Developers often work in pairs or small teams (e.g. **pair programming**)
5. **Test continuously**
    1. Perform **unit testing**, **integration testing**, and **acceptance testing** during the sprint
    2. Testing is **ongoing**, not saved for the end
6. **Review progress (Sprint Review)**
    1. Demo the working software to stakeholders
    2. Collect feedback and identify changes or improvements
7. **Reflect on process (Sprint Retrospective)**
    1. The team reflects on what went well and what needs improving in the next sprint
    2. Focus is on **team performance and process optimisation**
8. **Release** (may happen after every sprint or set of sprints)
    1. Deploy working software to users or staging environment
9. **Repeat**
    1. Move to the next sprint with updated priorities and feedback



Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

Benefits and drawbacks

Benefits	Drawbacks
Highly adaptable – responds quickly to changing requirements	**Requires experienced team members** – may be hard to manage without expertise
Frequent communication – promotes constant collaboration	**Risk of burnout** – intense collaboration can tire the team
Focus on quality – encourages good design and continuous testing	**May lack documentation** – flexibility can reduce written records
Customer collaboration – ensures the product meets real needs	**Scope creep** – changing goals may lead to uncontrolled growth



Your notes

Suitability

- The iterative model is most suitable for **small to medium-sized projects** where **requirements can change** and customer involvement is high

Rapid Application Development (RAD)

What is RAD?

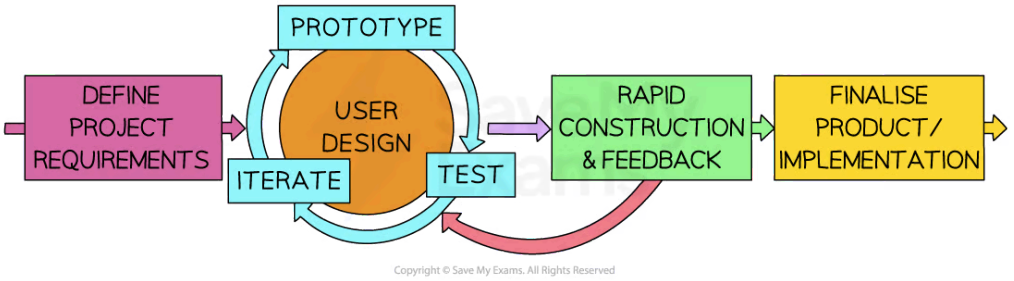
- Rapid Application Development (**RAD**) is a software development methodology that **emphasises fast and iterative development**

Steps

1. **Requirement planning:** Gather general system requirements, define constraints and assumptions
2. **User design and prototyping:** Collaborate with users to develop prototypes, ensuring alignment with user needs
3. **Construction or iterative development:** Build the system incrementally, with continuous user feedback and adaptation
4. **Cutover or deployment:** Transition the product into the live environment, including user training, support, and documentation
5. **Maintenance and updates:** Continue to adapt and improve the system based on user feedback and needs

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com



Rapid Application Development (RAD) Model of Software Development

Benefits and drawbacks

Benefits	Drawbacks
Speed – fast development and delivery at relatively low cost	**Requires strong team collaboration** – skilled and cohesive teams are essential
User involvement – client feedback shapes the system throughout	**Potential quality issues** – speed may reduce testing and documentation
Flexibility – adapts quickly to changing requirements	**Not ideal for small projects** – may be too complex for simple systems
Incremental development – builds in small, testable parts	**Scope creep risk** – flexibility can lead to ever-expanding requirements

Suitability

- Rapid Application Development is most suitable for projects where **rapid delivery is required** and where requirements can be developed and **refined on the go**

Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Program Design

Contents

- Structure charts
- State-Transition Diagrams

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Structure charts

What is a structure chart?

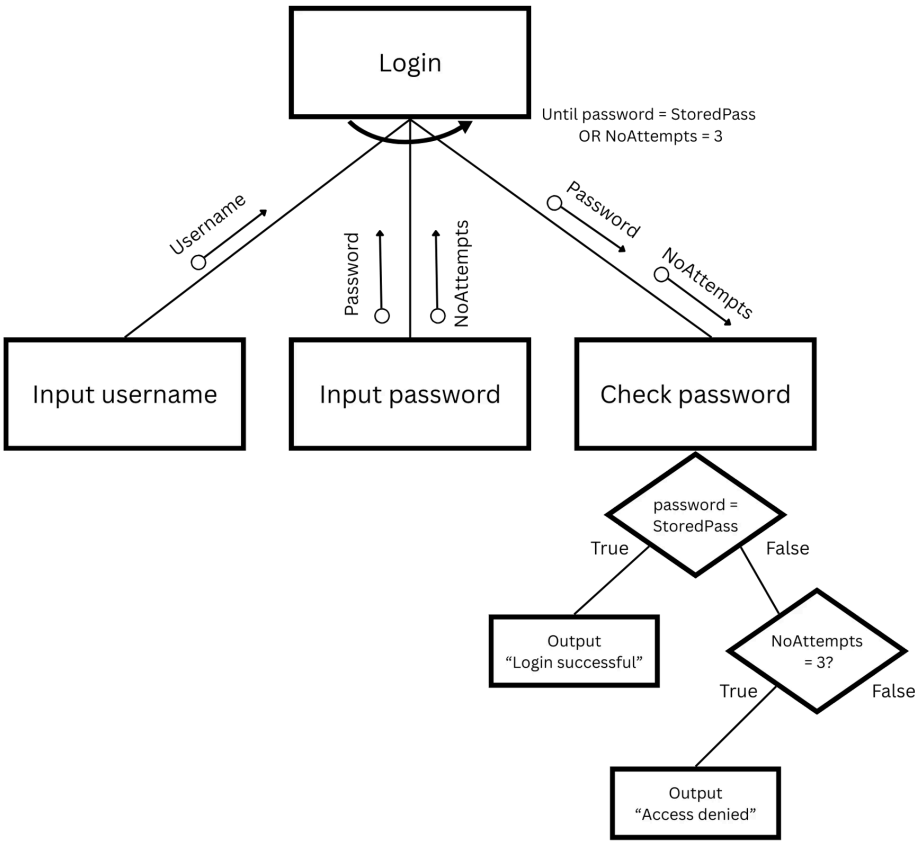
- A structure chart is a **modelling tool** used in the **design stage** of program development
- It helps to **decompose a problem** into smaller, manageable **sub-tasks**, representing each as a **module**
- Structure charts focus on **what the program does**, not how it does it

Example

- A program that asks a user to enter their username and password
- Checks if the password matches the stored password and uses a counter to track how many times it is checked
- If the password matches, a successful message is displayed, else after three failed attempts a 'denied' message is displayed
- A structure chart might look like:

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com



Your notes

Key features

Feature	Explanation
Top-down design	The chart starts with the **main program** and breaks it into sub-modules
Decomposition	Each module represents a **specific task** or **function**
Module boxes	Each module is shown as a **box**
Vertical lines	Show **control flow** – which module calls which
Parameter arrows	**Arrows** pointing into modules show **data being passed in or returned**

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Stepwise refinement	Each level adds **more detail** to the task above
Diamond shape	Shows a **condition** that could be **true or false**
Semi-circular arrow	Indicates **repetition**



Your notes

Pseudocode from structure chart

- The first step is to create an **identifier table**

Identifier	Data type	Description
Username	STRING	Stores the username entered by the user
Password	STRING	Stores the password entered by the user
StoredPass	STRING	The correct password stored in the system for comparison
NoAttempts	INTEGER	Counts the number of failed login attempts
LoginSuccess	BOOLEAN	Indicates whether the login was successful

- Next, identify is any **functions** or **procedures** could be used

Module	Type	Purpose
GetCredentials	**PROCEDURE**	Asks the user to input username and password
CheckPassword	**FUNCTION**	Compares input password with stored password and returns TRUE/FALSE
ShowAccessMessage	**PROCEDURE**	Displays success or failure message

- Any finally, write the pseudocode

```
DECLARE Username : STRING
DECLARE Password : STRING
DECLARE StoredPass : STRING
DECLARE NoAttempts : INTEGER
DECLARE LoginSuccess : BOOLEAN

StoredPass ← "admin123"
NoAttempts ← 0
LoginSuccess ← FALSE
```


© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

```
PROCEDURE GetCredentials()
   OUTPUT "Enter username: "
   INPUT Username
   OUTPUT "Enter password: "
   INPUT Password
ENDPROCEDURE

FUNCTION CheckPassword(P : STRING) RETURNS BOOLEAN
   IF P = StoredPass THEN
      RETURN TRUE
   ELSE
      RETURN FALSE
   ENDIF
ENDFUNCTION

PROCEDURE ShowAccessMessage(Success : BOOLEAN)
   IF Success = TRUE THEN
      OUTPUT "Login successful"
   ELSE
      OUTPUT "Access denied"
   ENDIF
ENDPROCEDURE

WHILE NoAttempts < 3 AND LoginSuccess = FALSE
   CALL GetCredentials()
   LoginSuccess ← CheckPassword(Password)

   IF LoginSuccess = FALSE THEN
      OUTPUT "Incorrect password"
      NoAttempts ← NoAttempts + 1
   ENDIF
ENDWHILE

CALL ShowAccessMessage(LoginSuccess)
```



Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

State-Transition Diagrams

What are state-transition diagrams?

- A state-transition diagram shows **how a system moves between different states** based on inputs
- Each **state** represents a condition or situation of the system
- **Transitions** are arrows that show movement between states when certain inputs occur
- Transitions may also produce **outputs**, shown as labels on the arrows
- These diagrams model **finite-state machines (FSMs)**, which are widely used in computing

Key elements

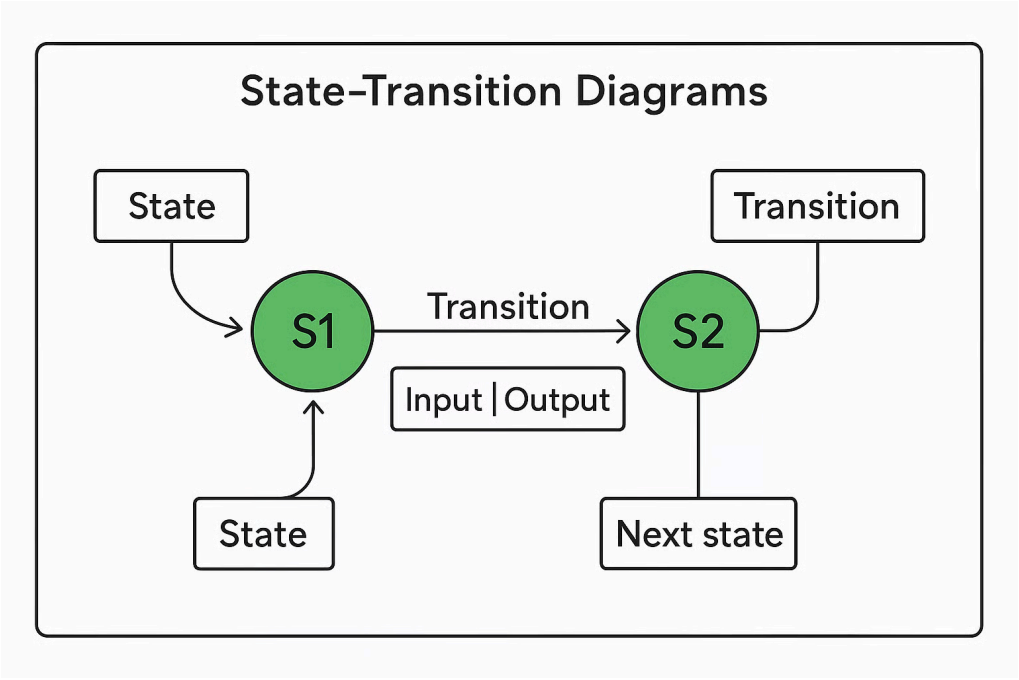
Element	Description
State	Circle labelled with a name (e.g. S1, S2) showing a situation the system can be in
Initial state	Starting point, sometimes marked with an arrow pointing to it
Transition	Arrow from one state to another, labelled with input and output
Input	Event or data that causes a transition (e.g. Button-Y pressed)
Output	Action or signal produced by a transition (e.g. Output-A)
Next state	The state the system moves into after the transition

How to read labels on transitions

- Labels are often written as **Input | Output**
- If no output is produced, "none" may be written
- Example: Button-Y | Output-B means pressing Button-Y in the current state produces Output-B and causes a transition to the next state
- The diagram below shows the key features of a state-transition diagram:

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com



Your notes

- States are circles labelled S1, S2, etc
- Transitions are arrows between states, labelled with Input | Output
- The initial state is marked with an incoming arrow

Completing transition tables

- Exam questions often ask you to convert a diagram into a **table**
- The table normally has columns for **Current state, Input, Output, Next state**
- Work through the diagram row by row, following the arrows for each possible input

Example table structure:

Current state	Input	Output	Next state
S1	Button-Y	Output-A	S2
S1	Button-Z	Output-B	S3

Examiner Tips and Tricks

- Always identify the **initial state** before tracing
- Highlight or mark inputs and outputs on your diagram to avoid missing one
- For tables, keep the order consistent: Input → Output → Next state
- Use arrows systematically, avoid guessing paths

© 2026 Save My Exams, Ltd.

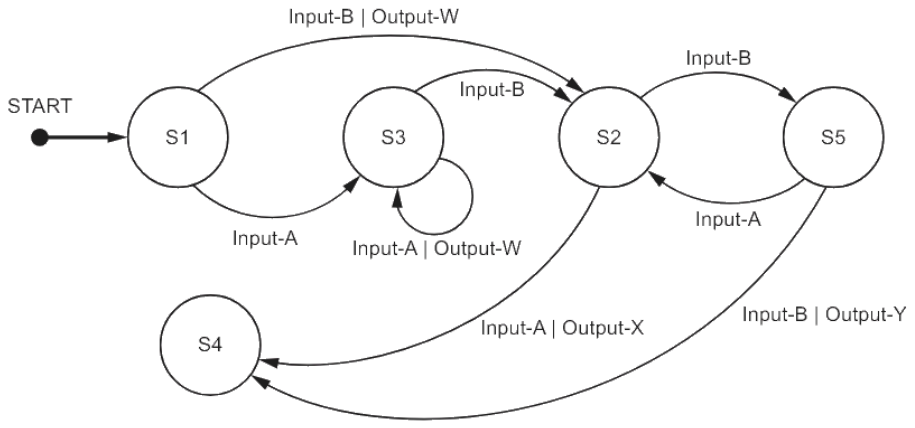
Study copy distributed by EduResource ZA | eduresourcza.com

- Remember: more than one sequence can sometimes reach the same state, but exams often ask for the **minimum changes**

Your notes

Worked Example

Part of a program is represented by the following state-transition diagram



A. Complete the table to show the inputs, outputs and next states.

Assume that the current state for each row is given by the 'Next state' on the previous row. For example, the first Input-A is made when in state S1.

If there is no output for a given transition, then the output cell should contain 'none'.

The first two rows have been completed.

Input	Output	Next state
		S1
Input-A	none	S3
	Output-W	
	none	
Input-B		
Input-A		
		S4

[5]

B. Identify the input sequence that will cause the minimum number of state changes in the transition from S1 to S4

© 2026 Save My Exams, Ltd. | Get more and ace your exams at savemyexams.com

Study copy distributed by EduResource ZA | eduresourcza.com

[1]

Your notes

Answers

A.

Input	Output	Next state
		S1
Input-A	none	S3
Input-A	Output-W	S3
Input-B	none	S2
Input-B	none	S5
Input-A	none	S2
Input-A	Output-X	S4

- One mark per row 3 to 7 [5 marks]

B. Input-B, Input-A [1 mark]

Study copy distributed by EduResource ZA | eduresourcza.com

Cambridge (CIE) AS Computer Science

Your notes

Program Testing & Maintenance

Contents

- Faults & errors
- Testing methods
- Maintenance

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Error types

What are the common types of error?

- Designing algorithms is a skill that must be developed and when designing algorithms, **mistakes and issues will occur**
- **Trace tables** can also help to find any kind of error in a program or algorithm
- There are three main categories of errors that when designing algorithms a programmer must be able to identify & fix, they are:
  - **Syntax errors**
  - **Logic errors**
  - **Runtime errors**

Syntax errors

What is a syntax error?

- A syntax error is **an error that breaks the grammatical rules of a programming language** and **stops it from running**
- Examples of syntax errors are:
  - **Typos and spelling errors**
  - **Missing or extra brackets or quotes**
  - **Misplaced or missing semicolons**
  - **Invalid variable or function names**
  - **Incorrect use of operators**
  - **Incorrectly nested loops & blocks of code**

Pseudocode – with syntax errors

```
FUNCTION generate_username(first_name STRING, last_name STRING) RETURNS STRING
   DECLARE username : STRING
   username = SUBSTRING(first_name, 1, 1) + SUBSTRING(last_name, 1, 1) + SUBSTRING(last_name, 1, 3)
   RETURN username
ENDFUNCTION

PROCEDURE main()
   DECLARE first_name : STRING
   DECLARE last_name : STRING
   DECLARE username : STRING
```


© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

```
    OUTPUT "Enter your first name: "
    INPUT first_name

    OUTPUT "Enter your last name: "
    INPUT last_name

    username ← generate_username[first_name, last_name]

    OUTPUT "Here's a suggested username:" + username
ENDPROCEDURE

// Main program
CALL Main
```

Pseudocode – without syntax errors

```
FUNCTION generate_username(first_name : STRING, last_name : STRING) RETURNS STRING
    DECLARE username : STRING
    username ← SUBSTRING(first_name, 1, 1) + SUBSTRING(last_name, 1, 1) + SUBSTRING(last_name, 1, 3)
    RETURN username
ENDFUNCTION

PROCEDURE main()
    DECLARE first_name : STRING
    DECLARE last_name : STRING
    DECLARE username : STRING

    OUTPUT "Enter your first name: "
    INPUT first_name

    OUTPUT "Enter your last name: "
    INPUT last_name

    username ← generate_username(first_name, last_name)

    OUTPUT "Here's a suggested username: ", username
ENDPROCEDURE

// Main program
CALL main()
```

Syntax errors

- FUNCTION generate_username(first_name STRING...)
  - **Missing colon** : between parameter name and type
  - first_name : STRING and last_name : STRING
- username = SUBSTRING(...)
  - **Uses =** instead of correct assignment operator ←
  - Should be username ← ...
- username ← generate_username[first_name, last_name]

Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

- **Incorrect use of square brackets []** instead of parentheses () for function call
- Should be `generate_username(first_name, last_name)`

- `CALL Main`
  - **Incorrect capitalisation and missing parentheses** — procedure name is case-sensitive
  - Should be `CALL main()`



Your notes

Logic errors

What is a logic error?

- A logic error is where **incorrect code is used that causes the program to run**, but **produces an incorrect output or result**
- Logic errors can be difficult to identify by the person who wrote the program, so one method of finding them is to use '**Trace Tables**'
- Examples of logic errors are:
  - **Incorrect use of operators (< and >)**
  - **Logical operator confusion (AND for OR)**
  - **Looping one extra time**
  - **Indexing arrays incorrectly (arrays indexing starts from 0)**
  - **Using variables before they are assigned**
  - **Infinite loops**

Pseudocode

```
FUNCTION calculate_area(length : REAL, width : REAL) RETURNS REAL
  // Checks for valid dimensions and calculates area
  IF length > 0 OR width > 0 THEN
    DECLARE area : REAL
    area ← width - length
    RETURN area
  ELSE
    OUTPUT "Length and width must be positive values."
  ENDIF
ENDFUNCTION

PROCEDURE main()
  DECLARE length : REAL
  DECLARE width : REAL
  DECLARE area : REAL

  OUTPUT "Enter the length of the rectangle: "
  INPUT length
```


Study copy distributed by EduResource ZA | eduresourcza.com

```
    OUTPUT "Enter the width of the rectangle: "
    INPUT width

    area ← calculate_area(length, width)

    OUTPUT "The area of the rectangle is approximately ", area, " square units."
ENDPROCEDURE

// Main program
CALL main()
```



Your notes

Logic errors

Line	Error	Correction
IF length > 0 OR width > 0 THEN	The check allows one value to be negative, which is not valid	Should be IF length > 0 AND width > 0 THEN
area ← width - length	Incorrect calculation – subtracts instead of multiplying to find the area	Should be area ← length * width

Runtime errors

What is a runtime error?

- A runtime error is where **an error causes a program to crash**
- Examples of runtime errors are:
  - **Dividing a number by 0**
  - **Index out of the range of an array**
  - **Unable to read or write a drive**

Pseudocode

```
FUNCTION calculate_area(length : REAL, width : REAL) RETURNS REAL
    // Calculates area of a rectangle after input validation
    IF length < 0 OR width < 0 THEN
        OUTPUT "Length and width must be positive values."
    ENDIF

    DECLARE area : REAL
    area ← length * width
    RETURN area
ENDFUNCTION

PROCEDURE main()
    DECLARE length : REAL
    DECLARE width : REAL
    DECLARE area : REAL
```


© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

```
    OUTPUT "Enter the length of the rectangle: "
    INPUT length

    OUTPUT "Enter the width of the rectangle: "
    INPUT width

    area ← calculate_area(length, width)

    OUTPUT "The area of the rectangle is approximately ", area, " square units."
ENDPROCEDURE

// Main program
CALL main()
```

Runtime error

Line	Error	Correction
area ← calculate_area(length)	Only one parameter is passed instead of two – will crash at runtime	Should be area ← calculate_area(length, width)

Worked Example

Each pseudocode statement in the following table may contain an error due to the incorrect use of the function or operator.

Describe the error in each case, or write 'NO ERROR' if the statement contains no error.

You can assume that none of the variables referenced are of an incorrect type. **[5]**

Statement	Error
Result ← 2 & 4	
SubString ← MID("pseudocode", 4, 1)	
IF x = 3 OR 4 THEN	
Result ← Status AND INT(x/2)	
Message ← "Done" + LENGTH(MyString)	

Answer

Statement	Error
Result ← 2 & 4	Should be arithmetic

Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

	operator (not &) **OR** 2 and 4 should be CHAR / STRING
SubString ← MID("pseudocode", 4, 1)	NO ERROR
IF x = 3 OR 4 THEN	Not Boolean values / incorrect operator **OR** Condition incorrect
Result ← Status AND INT(x/2)	INT(X/2) doesn't evaluate to a Boolean value / incorrect operator
Message ← "Done" + LENGTH(MyString)	Can't add string to number / "Done" is not a number



Your notes

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Testing methods

What are the different methods of testing?

- Testing takes place during the **testing stage** of the **program development life cycle**, once the program has been written and compiled
  - **Syntax errors** are usually detected during compilation
  - Testing focuses on checking for **logic errors**, **unexpected behaviour**, and **operational integrity**
  - A detailed **test plan**, outlining test cases and expected results, should be created during the **analysis stage**

Sample test plan

- Scenario: A program accepts a user's age between 0 and 120 inclusive

Test No.	Description	Input	Expected output	Type of test data	Actual outcome
1	Valid age within accepted range	25	"Age accepted"	Normal	
2	Input is not a number	"twenty"	"Invalid input: enter a number"	Abnormal	
3	Age above the maximum allowed	130	"Age out of range"	Extreme	
4	Age exactly at the upper boundary	120	"Age accepted"	Boundary	
5	Age exactly at the lower boundary	0	"Age accepted"	Boundary	
6	Age just below lower boundary	-1	"Age out of range"	Extreme	
7	Blank input	(blank)	"Please enter your age"	Abnormal	

Common methods of testing

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Testing method	Description
Dry run	**Manually trace through** the code (e.g. on paper) to predict output and track variables
Walkthrough	Step-by-step review of the code **with others** to identify issues early
White-box testing	Tests the **internal logic** and code structure; the tester knows how the program works
Black-box testing	Tests the **inputs and expected outputs** without knowing the internal workings
Integration testing	Checks that different modules or components **work correctly together**
Alpha testing	Performed **in-house** by the developers during early testing
Beta testing	Carried out by **external users** in real-world environments
Acceptance testing	Final check to ensure the program meets the original **client requirements**
Stub testing	Uses temporary modules (stubs) to **simulate missing components** during early testing



Your notes

Choosing suitable test data

What is suitable test data?

- Suitable test data is **specially chosen to test the functionality** of a program or design
- Developers or test-users would pick **a selection of test data** from the following categories
  - **Normal**
  - **Abnormal**
  - **Extreme**
  - **Boundary**
- The results would be **compared to the expected results** to check if the algorithm/program works as intended
- The results would be **stored in the test plan**
- Each category is explained within the context of a simple pseudocode program below

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Pseudocode
DECLARE name : STRING DECLARE age : INTEGER OUTPUT "What is your name? " INPUT name OUTPUT "How old are you? " INPUT age IF age >= 12 AND age <= 18 THEN   OUTPUT "Welcome, " + name + "! Your age is accepted." ELSE   OUTPUT "Sorry, " + name + ". Your age is not accepted." ENDIF



Your notes

Normal data

- Normal test data is data that **should be accepted** in the program
- An example would be a user entering their age as 16 into the age field of the program

Abnormal data

- Abnormal test data is data that is the **wrong data type**
- An example would be a user entering their age as "F" into the age field of the program

Extreme data

- Extreme test data is the **maximum** and **minimum** values of normal data that are **accepted** by the system
- An example would be a user entering their age as 18 or 12 into the age field of the program

Boundary data

- Boundary test data is similar to extreme data except that the values **on either side of** the **maximum** and **minimum** values are tested
- The largest and smallest **unacceptable values**
- An example would be a user entering their age as 11 or 19 into the age field of the program

Selecting suitable test data

Type of Test	Input	Expected Output
Normal	14	Accepted
Normal	16	Accepted

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com

Extreme	12	Accepted
Extreme	18	Accepted
Abnormal	H	Rejected
Abnormal	@	Rejected
Boundary	11	Rejected
Boundary	19	Rejected



Your notes

Study copy distributed by EduResource ZA | eduresourcza.com

Your notes

Types of maintenance

What is program maintenance?

- Program maintenance is the process of updating or improving a program **after it has been delivered** to the user
- Unlike physical equipment, programs **don't wear out**, but they may need to be changed due to **errors**, **changing requirements**, or **new technology**

Why maintenance is needed

- To **fix errors that were missed** during testing
- To **improve performance** based on user feedback
- To **adapt the software** to new uses or platforms

Types of program maintenance

Type	Purpose	Example
Corrective	Fixes bugs or errors found during real-world use	Fixing a bug that causes the program to crash when special characters are entered
Perfective	Improves performance or adds small enhancements	Replacing a loading screen with a progress bar to give better user feedback
Adaptive	Modifies the program to support new environments or requirements	Modifying the program to work on a tablet instead of just a desktop computer

© 2026 Save My Exams, Ltd.

Study copy distributed by EduResource ZA | eduresourcza.com