TM111 + TM112
Ultimate Study Companion

All the essential knowledge you need for TM111 & TM112

Paul's Edition

Mini Glossary

A fast reference list of some of the most important terms you will see in TM111 and TM112.

Algorithm: A step-by-step method for solving a problem.

Binary: A base-2 number system using only 0 and 1.

Bit: The smallest unit of data (0 or 1).

Byte: A group of 8 bits.

CPU: Central Processing Unit, executes instructions.

Data: Information stored or processed by a computer.

Database: A structured collection of related data.

Flowchart: A diagram that shows the steps in an algorithm.

Hexadecimal: Base-16 number system used to shorten binary.

HTML: Markup language used to structure web pages.

IP Address: A numeric label that identifies a device on a network.

Logic: Rules used for decision making in computing.

Network: Two or more devices connected to share data.

RAM: Fast temporary memory used by running programs.

Storage: Long-term saving of data such as SSD or HDD.

URL: Address used to locate resources on the web.

Web Browser: Software used to view web pages.

Table of Contents

Chapter 1: Maths & Core Formulas

Percentages

Increase: New = Original × (1 + percent/100)

Decrease: New = Original × (1 – percent/100)

Example: 200 increased by 15% → 200 × 1.15 = 230

Binary Basics

Binary uses base 2. Each position doubles:

128 64 32 16 8 4 2 1

Example: 10110101 = 128 + 32 + 16 + 4 + 1 = 181

Binary → Decimal

Add all place values that have a 1.

Example: 11001010 = 128 + 64 + 8 + 2 = 202

Decimal → Binary

Keep subtracting the largest power of 2.

Example: 45 → 32 + 8 + 4 + 1 → 101101

Hexadecimal

Hex is base 16 with digits 0–9 and A–F.

Hex → Decimal: multiply each digit by 16^position.

Example: 3A = (3×16) + 10 = 58

Decimal → Hex: divide by 16 and track remainders.

Memory Units

1 byte = 8 bits

1 KB = 1024 bytes

1 MB = 1024 KB

1 GB = 1024 MB

1 TB = 1024 GB

Image Size Formula

File Size = Width × Height × Bit Depth (bits) / 8

Example: 1920×1080 at 24-bit:

1920 × 1080 × 24 ÷ 8 = 6,220,800 bytes ≈ 6.22 MB

Colour Depth

Bit depth determines number of colours:

8-bit → 256 colours

16-bit → 65,536 colours

24-bit → 16.7 million colours

Pixel & Resolution

Total Pixels: width × height

Example: 2560×1440 = 3,686,400 pixels

Download Time Formula

Time = File Size (MB) / Speed (MB/s)

Convert Mbps: Mbps ÷ 8 = MB/s

File Size Formula

File Size (bytes) = bits ÷ 8

Media: sample_rate × bit_depth × channels × seconds

Chapter 2: Data & Storage

What is Data?

Data is information represented in a form computers can process.

Computers store and transmit data using binary (0s and 1s).

Bits, Bytes & Encoding

Bit = smallest unit of data (0 or 1).

Byte = 8 bits.

Encodings allow meaning to be assigned to binary patterns.

Example: 01000001 represents 'A' in ASCII.

ASCII

ASCII uses 7 bits to represent English characters.

128 possible characters (0–127).

Unicode

Unicode supports characters from all languages.

UTF■8 is the most common encoding; uses 1–4 bytes per character.

Data Formats

Text files store characters using ASCII or Unicode.

Images store pixels and colour information.

Audio stores samples of sound waves.

Video stores frames of images plus audio.

Compression

Lossless: reduces size without losing information. (ZIP, PNG, FLAC)

Lossy: removes some data to reduce size further. (JPEG, MP3, MP4)

Metadata

Metadata describes data.

Examples: image size, camera model, GPS location, author name.

Pixels & Resolution

Images are made of tiny squares called pixels.

Resolution = width × height.

Example: 1920×1080 = 2,073,600 pixels.

Bit Depth (Colour Depth)

Higher bit depth = more colours.

8-bit = 256 colours.

24-bit = 16.7 million colours (standard RGB).

Image File Size Formula

File Size = width × height × bit depth (bits) ÷ 8.

Example: 800×600 at 24-bit → 800×600×24 ÷ 8 = 1,440,000 bytes ≈ 1.44 MB.

Audio Data

Audio is stored by sampling sound waves.

Sample rate: number of samples per second (Hz).

Bit depth: amount of data per sample.

Channels: mono = 1, stereo = 2.

Audio size = sample_rate × bit_depth × channels × seconds ÷ 8.

Memory Types

RAM: fast temporary working memory.

ROM: permanent memory storing firmware.

Cache: small, very fast memory close to CPU.

Virtual Memory: uses storage as extension of RAM.

Storage Types

HDD: magnetic, slower, cheaper, moving parts.

SSD: fast, silent, no moving parts.

Flash Storage: used in USB drives, phones, SD cards.

Structured vs Unstructured Data

Structured: organised into tables (e.g., databases).

Unstructured: free-form data like images, videos, emails.

Analogue vs Digital

Analogue: continuous signals (sound, light).

Digital: represented using binary.

Sampling converts analogue signals into digital form.

Chapter 3: Networking & The Web

What is a Network?

A network is two or more devices connected to share data.

Networks allow devices to communicate and share resources.

Types of Networks

LAN: Local network in one building.

WAN: Wide area network across cities or countries.

PAN: Personal network (Bluetooth devices).

MAN: Metropolitan area network (city-wide).

Star Topology

```
+--------+ | Switch | +---+----+ +------+ +------+ | PC1 | | PC2 |
+------+ +------+ | | +------+ +------+ | PC3 | | PC4 | +------+ +------+
```

Bus Topology

```
+---------+ +---------+ +---------+ | PC1 |----| PC2 |----| PC3 |
+---------+ +---------+ +---------+ (single shared cable)
```

Wired vs Wireless

Wired: faster, stable, uses Ethernet cables.

Wireless: uses radio waves (Wi-Fi), more flexible but can be less stable.

Network Devices

Router: directs traffic between networks and provides IP addresses.

Switch: connects devices inside a LAN.

Access Point: provides Wi-Fi coverage.

Modem: connects home network to the internet provider.

IP Addresses

IPv4 format: 192.168.1.10 (four numbers 0–255).

IPv6 is the newer, longer format.

IP identifies each device uniquely on a network.

DNS Lookup (Simplified)

```
+--------+ +---------+ +----------+ | You | ----> | DNS | ----> |
Webserver| +--------+ +---------+ +----------+ Enter URL Returns IP Sends
website
```

Packets & Routing

Data travels in small chunks called packets.

Routers forward packets along the best path to their destination.

Bandwidth vs Latency

Bandwidth: how much data can be transferred (speed).

Latency: how long it takes for a signal to travel (delay).

Web vs Internet

The **Internet** is the physical network of cables and routers.

The **Web** is the collection of websites and services running on the internet.

URLs

Format: protocol://domain/path

Example: https://open.ac.uk/study

HTTP & HTTPS

HTTP transfers webpage data.

HTTPS encrypts the data for security.

HTML Basics

HTML structures a webpage.

<h1> heading, **<p>** paragraph, **<a>** link.

CSS Basics

CSS controls style: colours, layout, fonts.

SEO Checklist

Use meaningful page titles.

Use alt text for images.

Use headings correctly.

Keep pages fast and mobile-friendly.

Accessibility Basics

Use alt text.

Ensure good colour contrast.

Use proper headings and labels.

Chapter 4: Databases & Big Data

What is a Database?

A database is an organised collection of data stored electronically.

Allows fast searching, sorting, filtering and updating.

Entities & Attributes

Entity: thing being stored (e.g., Student).

Attributes: details about the entity (e.g., Name, ID, Course).

Tables

Tables store data in rows and columns.

Row = record, Column = attribute.

Primary Keys

A primary key uniquely identifies each record.

Example: StudentID.

Foreign Keys

A foreign key links one table to another.

Example: CourseID inside a Students table.

Relationships

One-to-One: one record links to one record.

One-to-Many: most common relationship.

Many-to-Many: needs a junction table.

Structured vs Unstructured Data

Structured: organised, tables, rows, columns (SQL databases).

Unstructured: images, video, audio, emails, documents.

SQL Basics

SELECT: get data.

WHERE: filter data.

INSERT: add data.

UPDATE: modify data.

DELETE: remove data.

Sampling

Used to analyse big datasets without processing everything.

Random sampling gives a representative subset.

Big Data Characteristics

Volume: huge data sizes.

Velocity: data created fast.

Variety: many formats.

Veracity: data reliability.

Distributed Systems

Large datasets stored across many machines.

Improves speed, reliability and capacity.

MapReduce (Simplified)

Map: split data into chunks and process in parallel.

Reduce: combine results.

Used by systems like Hadoop.

Chapter 5: Programming & Logic

What is an Algorithm?

An algorithm is a step-by-step set of instructions for solving a problem.

Algorithms must be clear, unambiguous, and finite.

Pseudocode

Pseudocode uses plain-English instructions to describe algorithms.

Example: IF age > 18 THEN print 'Adult'.

Flowcharts

Flowcharts use shapes to show algorithm steps.

Oval = Start/End, Rectangle = Process, Diamond = Decision.

Sequence, Selection, Iteration

Sequence: steps executed in order.

Selection: decisions using IF/ELSE.

Iteration: loops repeating actions (FOR, WHILE).

Boolean Logic

Boolean values: True or False.

AND: both must be true.

OR: at least one must be true.

NOT: reverses True/False.

Logical Operators

== equal, != not equal, > greater than, < less than.

>= greater or equal, <= less or equal.

Python Basics

Python uses indentation to show blocks.

Example: if x > 10: print('Large').

Variables & Data Types

Variables store data: x = 5.

int: whole numbers.

float: decimals.

str: text.

bool: True/False.

Input & Output

Input: name = input('Enter name: ')

Output: print('Hello', name)

If / Else in Python

if x > 5:

print('Big')

else:

print('Small')

Loops in Python

FOR: for i in range(5): print(i)

WHILE: while x < 10: x += 1

Functions

Functions group reusable code.

Example:

def greet():

print('Hello!')

Common TM111 Python Patterns

Counting loops.

Searching lists.

Processing strings.

Validating input using loops + conditions.

Chapter 6: AI, Machine Learning & Automation

What is Artificial Intelligence?

AI refers to systems that can perform tasks normally requiring human intelligence.

Examples: recognising speech, classifying images, translating languages.

Machine Learning Basics

Machine Learning (ML) is a subset of AI.

ML systems learn patterns from data rather than being explicitly programmed.

Training data is used to teach the model what patterns look like.

Training Data

Training data must be accurate and representative.

Poor data leads to poor model performance.

Large datasets help ML systems learn reliably.

Types of Machine Learning

Supervised: learns from labelled data (e.g., emails marked spam/not spam).

Unsupervised: finds hidden patterns (e.g., clustering customers).

Reinforcement: learns by trial and error (e.g., game-playing AI).

Neural Networks

Inspired by the human brain.

Useful for complex tasks like image recognition, speech, translation.

AI Examples

OCR: reading text from images.

Speech Recognition: converting speech to text.

Face Recognition: identifying people in photos.

Recommendation Systems: suggesting videos, music, products.

Automation

Automation uses technology to complete tasks with minimal human involvement.

Examples: factory robots, automated checkouts, software scripts.

Bias in AI

AI systems can learn human biases from training data.

If data contains unfair patterns, the AI may reproduce them.

Diverse and carefully curated datasets reduce bias.

Ethics & Responsibility

AI must be designed and used responsibly.

Important issues: privacy, fairness, accountability, transparency.

Users should understand how data is collected and used.

Limitations of AI

AI does not understand meaning; it recognises patterns.

AI can make mistakes when data is ambiguous or unfamiliar.

AI cannot reason or feel emotions like humans.

Chapter 7: Design Thinking & The Web

What is Design Thinking?

A creative, user-focused approach to solving problems.

Used to design products, websites, services, and systems.

The Design Thinking Cycle

1. Discover: explore the problem, gather information.

2. Define: identify user needs and the real challenge.

3. Develop: generate ideas and create possible solutions.

4. Deliver: test, refine, and complete the final design.

Naming, Framing, Moving, Reflecting

Naming: identify what the problem really is.

Framing: choose the lens or angle to approach it.

Moving: generate ideas, sketches, prototypes.

Reflecting: evaluate what works and what doesn't.

User-Centred Design

Design focuses on the needs, abilities and expectations of the user.

Good design solves real problems and reduces frustration.

Good Interface Design

Clear layout with consistent structure.

Readable text with good contrast.

Logical navigation.

Avoid clutter; keep important information obvious.

Use familiar icons and patterns.

Accessibility Principles

Provide alt text for images.

Ensure strong colour contrast.

Use headings properly.

Avoid relying solely on colour to show meaning.

Make content keyboard■friendly.

The Web vs The World Wide Web

The Internet = global network of computers.

The Web = collection of linked webpages accessed through browsers.

Weaving the Web (Key Ideas)

Tim Berners■Lee created the Web using three technologies:

1. HTML: page structure.

2. URLs: addresses for resources.

3. HTTP: protocol for transferring web content.

The Web works because it is open, link■based and decentralised.

Designing for the Web

Pages should load fast and display well on all devices.

Use meaningful headings and semantic HTML.

Keep content organised into sections.

Ensure consistent styling using CSS.

Full Glossary

Algorithm

A step-by-step method for solving a problem.

Analogue

Continuous signal that varies smoothly.

API

Interface that allows software to communicate.

ASCII

7-bit text encoding for English characters.

Attribute

A property describing an entity in a database.

Bandwidth

Amount of data transferred per second.

Binary

Base-2 number system using 0 and 1.

Bit

Smallest unit of data.

Bit Depth

Number of bits used per pixel or sample.

Boolean

Data type with True/False values.

Byte

8 bits.

Cache

Small, fast memory used by CPU.

Client

Device that requests data from a server.

Compression

Reducing file size (lossless or lossy).

CPU

Central Processing Unit, executes instructions.

CSS

Cascading Style Sheets for webpage styling.

Data

Information stored or processed by a computer.

Database

Organised collection of structured data.

Decimal

Base-10 number system.

DNS

System that converts domain names to IPs.

Download Time

Time needed to transfer a file based on speed.

Entity

Object represented in a database table.

Ethernet

Wired networking standard.

Flowchart

Diagram representing an algorithm.

Foreign Key

Attribute linking two tables together.

Function

Reusable block of code that performs a task.

GB

Gigabyte, 1024 MB.

GUI

Graphical User Interface.

Hexadecimal

Base-16 number system.

HTML

Markup language used to structure webpages.

HTTP

Protocol for transferring webpage data.

HTTPS

Secure, encrypted version of HTTP.

IP Address

Unique number identifying a device on a network.

Iteration

Repeating steps in a loop.

JPEG

Compressed image file format (lossy).

KB

Kilobyte, 1024 bytes.

LAN

Local Area Network.

Latency

Delay in data transmission.

Loop

Code that repeats while a condition holds.

Machine Learning

Systems that learn patterns from data.

MB

Megabyte, 1024 KB.

Metadata

Data describing other data.

Network

Two or more devices connected to share data.

OCR

Optical Character Recognition.

Packet

Small chunk of data sent across a network.

PAN

Personal Area Network.

Pixel

Smallest unit in a digital image.

Primary Key

Unique identifier for table records.

Protocol

Set of rules for data transmission.

RAM

Temporary memory used for active tasks.

ROM

Permanent memory used for firmware.

Router

Device directing traffic between networks.

Sampling

Turning analogue signals into digital values.

RGB

Colour model using red, green, blue values.

SQL

Language used for managing databases.

Storage

Long-term data saving (SSD, HDD).

Structured Data

Organised format like tables.

Unstructured Data

Free-form data like images or audio.

URL

Address used to locate a resource on the Web.

Variable

Named value stored in memory.

Wi-Fi

Wireless networking technology.

WWW

World Wide Web, collection of linked pages.

Chapter 8: Advanced Python

Lists

Lists store ordered data and can be changed (mutable).

Common methods: append, remove, sort, reverse.

```
my_list = [1, 2, 3] my_list.append(4) print(my_list)
```

List Slicing

Slices extract parts of a list using start:end syntax.

```
nums = [10,20,30,40] print(nums[1:3])
```

Dictionaries

Dictionaries store key-value pairs.

Keys must be unique.

```
person = {'name': 'Paul', 'age': 40} print(person['name'])
```

Tuples & Sets

Tuples: ordered, immutable lists.

Sets: unordered collections with no duplicates.

```
items = {'apple','banana','apple'} print(items)
```

String Processing

Strings support slicing, searching and replacing.

```
text = 'hello world' print(text.upper())
```

List Comprehensions

Short way to build lists.

```
squares = [x*x for x in range(10)]
```

Functions with Parameters

Functions take inputs and return outputs.

```
def add(a,b): return a+b
```

Returning Values

Use return to output results from a function.

```
def square(n): return n*n
```

Error Handling

Try/except prevents crashes.

```
try: x = int('abc') except: print('Error!')
```

File Handling

Open, read, and write text files.

```
with open('data.txt','r') as f: print(f.read())
```

Modules and Imports

Modules contain reusable Python code.

```
import math print(math.sqrt(25))
```

Chapter 9: Full HTML Reference

HTML Document Structure

All webpages follow a basic structure:

```
<!DOCTYPE html> <html> <head> <title>Page Title</title> </head> <body>
Content goes here </body> </html>
```

Head Section Tags

<title> – Sets the page title.

<meta> – Metadata such as charset or description.

<link> – Load CSS files.

<script> – Load JavaScript files.

Body Section Tags

<h1> to <h6> – Headings.

<p> – Paragraph.

** ** – Block-level container.

**** – Inline container.

Links & Images

<a href=""> – Creates a link.

<img src="" alt=""> – Displays an image.

Image **alt text** is required for accessibility.

Lists

<ul> – Unordered list.

<ol> – Ordered list.

<li> – List item.

```
<ul> <li>Item 1</li> <li>Item 2</li> </ul>
```

Tables

<table> – Table container.

<tr> – Table row.

<td> – Cell.

<th> – Header cell.

```
<table> <tr><th>Name</th><th>Age</th></tr>
<tr><td>Paul</td><td>40</td></tr> </table>
```

Forms & Inputs

<form> – Form container.

<input> – Text, password, email, number, file, etc.

<label> – Accessible label for an input.

<textarea> – Multi-line text.

<select> – Dropdown menu.

```
<form> <label>Name:</label> <input type="text"> </form>
```

Semantic HTML

<header> – Top of a page.

<nav> – Navigation menus.

<section> – Page sections.

<article> – Standalone content.

<footer> – Page footer.

Semantic HTML improves accessibility and SEO.

Media Tags

<audio> – Embed audio.

<video> – Embed video.

<source> – Define audio/video file source.

Attributes Reference

id – Unique identifier.

class – Used for CSS.

src – Image or media source.

href – Link target.

alt – Image description.

style – Inline CSS (avoid when possible).

Meta Tags (SEO)

<meta charset="UTF-8"> – Character encoding.

<meta name="description" content=""> – Page description.

<meta name="viewport" content="width=device-width, initial-scale=1"> – Mobile scaling.

Chapter 10: CSS Quick Reference

What is CSS?

CSS controls the layout, colour, fonts and overall appearance of a webpage.

It works alongside HTML which provides structure.

Basic Rule Structure

A CSS rule has a selector and declarations:

```
p { color: blue; font-size: 16px; }
```

Selectors

Element selectors: p, h1, ul

Class selectors: .menu, .title

ID selectors: #header

Universal selector: *

Group selectors: h1, h2, p

Colours

Colour formats include HEX, RGB, and names.

```
color: #ff0000; color: rgb(255,0,0);
```

Text Styling

Font size, weight, style, alignment:

```
p { font-size: 16px; font-weight: bold; text-align: center; }
```

Box Model

Every element is a box with content, padding, border and margin.

```
div { padding: 10px; margin: 20px; border: 1px solid black; }
```

Display Types

block: takes full width.

inline: stays in a line.

inline-block: inline but accepts width/height.

none: hidden.

Flexbox Basics

Flexbox is used to arrange items in rows or columns.

```
.container { display: flex; gap: 10px; }
```

Grid Basics

CSS Grid creates full layout grids.

```
.grid { display: grid; grid-template-columns: 1fr 1fr; gap: 20px; }
```

Positioning

static: default.

relative: moved relative to itself.

absolute: positioned relative to parent.

fixed: stays on screen.

```
div { position: absolute; top: 10px; left: 20px; }
```

Backgrounds

```
body { background-color: #f0f0f0; } body { background-image:
url('img.jpg'); }
```

Media Queries

Used for responsive design.

```
@media (max-width: 600px) { body { font-size: 14px; } }
```

CSS Best Practices

Keep CSS in external .css files.

Use classes instead of IDs for styling.

Keep naming consistent.

Avoid inline styles.

Group related rules together.

Chapter 11: SQL Cheatsheet

What is SQL?

SQL (Structured Query Language) is used to manage and interact with relational databases.

Used to search, filter, insert, update and delete data.

Basic SELECT

Get data from a table.

```
SELECT * FROM Students;

SELECT name, age FROM Students;
```

WHERE Clause

Filters rows by a condition.

```
SELECT * FROM Students WHERE age > 18;

SELECT * FROM Users WHERE active = 1;
```

ORDER BY

Sort results.

```
SELECT name, age FROM Students ORDER BY age DESC;
```

LIMIT

Restrict number of results.

```
SELECT * FROM Logs LIMIT 10;
```

INSERT

Add new data.

```
INSERT INTO Students (name, age) VALUES ('Paul', 40);
```

UPDATE

Modify existing data.

```
UPDATE Students SET age = 41 WHERE name = 'Paul';
```

DELETE

Remove rows.

```
DELETE FROM Students WHERE age < 18;
```

JOINs

JOIN combines data from multiple tables.

INNER JOIN

Returns matching records from both tables.

```
SELECT Students.name, Courses.title FROM Students INNER JOIN Courses ON
Students.courseID = Courses.id;
```

LEFT JOIN

Returns all records from left table even if no match.

```
SELECT * FROM Students LEFT JOIN Courses ON Students.courseID =
Courses.id;
```

GROUP BY & Aggregates

GROUP BY groups similar values.

Aggregates: COUNT(), AVG(), SUM(), MIN(), MAX().

```
SELECT courseID, COUNT(*) FROM Students GROUP BY courseID;
```

Primary & Foreign Keys

Primary Key: unique identifier for records.

Foreign Key: links tables together.

Normalisation Basics

Normalisation reduces duplication and improves consistency.

1NF: no repeating groups.

2NF: no partial dependencies.

3NF: no transitive dependencies.

Chapter 12: Web Developer Deep Dive

HTTP Status Codes

HTTP status codes tell the browser what happened to a request:

200 OK – Success.

301 Moved Permanently – Redirect.

400 Bad Request – Client error.

401 Unauthorized – Login required.

403 Forbidden – No permission.

404 Not Found – Missing page.

500 Internal Server Error – Server crashed.

DNS (Deep Explanation)

DNS converts human-friendly names to IP addresses.

Your browser asks DNS for the IP of a domain:

1. Browser → DNS Resolver

2. Resolver → Root Server

3. Root → TLD Server (.com, .net, .org)

4. TLD → Domain's Name Server

5. Name Server returns the IP address

Cookies

Small bits of data stored in your browser.

Used for login sessions, preferences, tracking.

Sent automatically with every request to that domain.

HTTP Headers

Headers contain important metadata.

Request headers: User-Agent, Accept, Cookie.

Response headers: Content-Type, Set-Cookie, Cache-Control.

HTTPS & TLS

HTTPS encrypts data so attackers cannot read it.

TLS provides encryption, authentication and integrity.

The padlock icon means a secure connection.

REST APIs

REST is a style for building web APIs.

Common HTTP methods:

GET: retrieve data.

POST: create data.

PUT: update data.

DELETE: remove data.

APIs return data in JSON format.

```
{ "name": "Paul", "age": 40 }
```

Caching

Caching improves speed by storing files temporarily:

Browser cache: stores images, CSS, JS.

Server cache: stores rendered pages.

CDN cache: stores files near users worldwide.

Content Delivery Networks (CDNs)

CDNs deliver images, videos and static files faster by using global servers.

Examples: Cloudflare, Akamai, Amazon CloudFront.

Reduces latency and speeds up websites.

Chapter 13: Operating System Fundamentals

What is an Operating System?

The OS manages hardware, software, memory, storage and user interfaces.

Examples include Windows, macOS, Linux, iOS and Android.

Processes & Threads

Process: a running program.

Thread: a smaller unit inside a process.

Multi-threading allows a process to perform multiple tasks at once.

Scheduling

The CPU can only run a few processes at a time.

Schedulers decide which process runs next.

Common scheduling methods include:

- First Come First Served (FCFS)

- Shortest Job First (SJF)

- Round Robin (RR)

Memory Management

The OS allocates RAM to processes.

Keeps programs separate so they do not overwrite each other.

Prevents crashes using memory protection.

Virtual Memory

When RAM is full, the OS uses storage as temporary memory.

This space is called the swap file or page file.

Allows more programs to run, but slower than real RAM.

File Systems

File systems organise how files are stored.

Examples: NTFS, FAT32, exFAT, EXT4.

They track folders, permissions, metadata and storage locations.

User Accounts

OS supports different types of accounts:

- Standard users

- Administrators

- Guests

User accounts control access to files and settings.

Permissions

Permissions define who can read, write or execute a file.

Linux: rwx permissions for owner, group and others.

Windows: NTFS permissions (Full, Modify, Read, Execute).

Storage Management

The OS manages HDDs, SSDs, USB drives and external storage.

Handles formatting, mounting, repairing and partitioning.

Keeps track of free space, bad sectors and file allocation.

Chapter 14: Maths Expansion

Binary Arithmetic

Binary addition works like decimal but carries at 2.

0+0=0, 1+0=1, 1+1=0 carry 1.

```
1011 + 0110 ------- 10001
```

Negative Binary (Two’s Complement)

To find negative: invert bits then add 1.

Example: -5 in 8-bit:

```
5 = 00000101 Invert = 11111010 Add 1 = 11111011
```

Hexadecimal Conversion

Hex uses digits 0–9 and A–F.

Group binary in 4s to convert.

```
11010110 = D6
```

Exponents

2^10 = 1024 (about 1 KB).

Computing uses powers of 2 heavily.

Logarithms in Computing

Base-2 logs show how many bits are needed.

Example: log2(256) = 8 bits.

Unit Conversions

1 byte = 8 bits.

1 KB = 1024 bytes.

1 MB = 1024 KB.

1 GB = 1024 MB.

Download Time Formula

Download time depends on file size and bandwidth.

Formula:

```
time = file_size (bits) / speed (bits per second)
```

Example: 100 MB file at 50 Mbps:

100 MB = 800 Mb → 800 ÷ 50 = 16 seconds.

Percentage Cheat Sheet

10% of X = X / 10.

20% of X = X × 0.2.

Increase by %: new = old × (1 + p/100).

Decrease by %: new = old × (1 - p/100).

Bit/Byte Conversion Table

8 bits = 1 byte

1024 bytes = 1 KB

1024 KB = 1 MB

1024 MB = 1 GB

1024 GB = 1 TB

Chapter 15: AI & Machine Learning (Advanced)

Training, Validation & Testing

Training set: used to teach the model patterns.

Validation set: used to tune settings (hyperparameters).

Test set: final evaluation to check real performance.

Hyperparameters

Settings chosen before training:

- Learning rate

- Batch size

- Number of layers

- Number of neurons

Hyperparameters affect accuracy and speed.

Neural Network Layers

Input layer: receives raw data.

Hidden layers: extract patterns.

Output layer: final prediction.

Deep networks contain many hidden layers.

Convolution Basics (Computer Vision)

Convolutional Neural Networks (CNNs) detect visual features.

Filters scan the image for edges, shapes, textures.

Deeper layers detect complex patterns like faces or objects.

NLP Basics (Tokenisation & Embeddings)

Tokenisation breaks text into pieces (words or subwords).

Embeddings convert words into numeric vectors.

Similar meanings → vectors close together.

Overfitting & Underfitting

Overfitting: model memorises training data, performs poorly on new data.

Underfitting: model is too simple and fails to learn patterns.

Goal: balance accuracy with generalisation.

Data Augmentation

Used to improve training data:

- Rotate images

- Flip images

- Add noise

- Change brightness

Helps reduce overfitting.

Model Evaluation Metrics

Accuracy: percentage of correct predictions.

Precision: correctness of positive predictions.

Recall: ability to find all positives.

F1 Score: balance of precision and recall.

Bias & Fairness

AI can inherit human biases from data.

Biased training → unfair results.

Fair models require balanced, representative datasets.

Risk & Responsible AI

Key principles:

- Transparency

- Privacy

- Accountability

- Safety

AI should be used responsibly to avoid harm.