Computer Fundamentals

Presented by: Adarsh Pandey

MCA 1st Year

Subject: Fundamentals of Computer

What is a Computer?

An electronic device that processes data according to software instructions. It takes input, processes it with a CPU, stores information, and produces output to perform various tasks.

Types of Computers

Desktops

Regular use, separate components (monitor, keyboard, mouse, CPU) mounted together.

Laptops

Portable, all components integrated, great for on-the-go work with built-in features.

Servers

Manage network resources, provide services like databases and application hosting.

Tablets

More portable than laptops, touchscreens for browsing, content, and communication.

Other devices include smartphones, game consoles, and Smart TVs.

Computers by Functionality & Size

By Functionality

Analog Computers

Data stored using continuous physical quantities (e.g., mechanical integrators).

Digital Computers

Most common, process data using discrete values (e.g., smartphones).

Hybrid Computers

Combine analog and digital (e.g., complex medical equipment).

By Size

Microcomputers

Small, compact, for individual use (e.g., smartphones, desktops).

Minicomputers

More powerful than microcomputers, used in mid-sized businesses (e.g., servers).

Mainframe Computers

Used by large organizations for bulk data processing.

Supercomputers

Extremely powerful for complex calculations, research purposes.

Hardware vs. Software

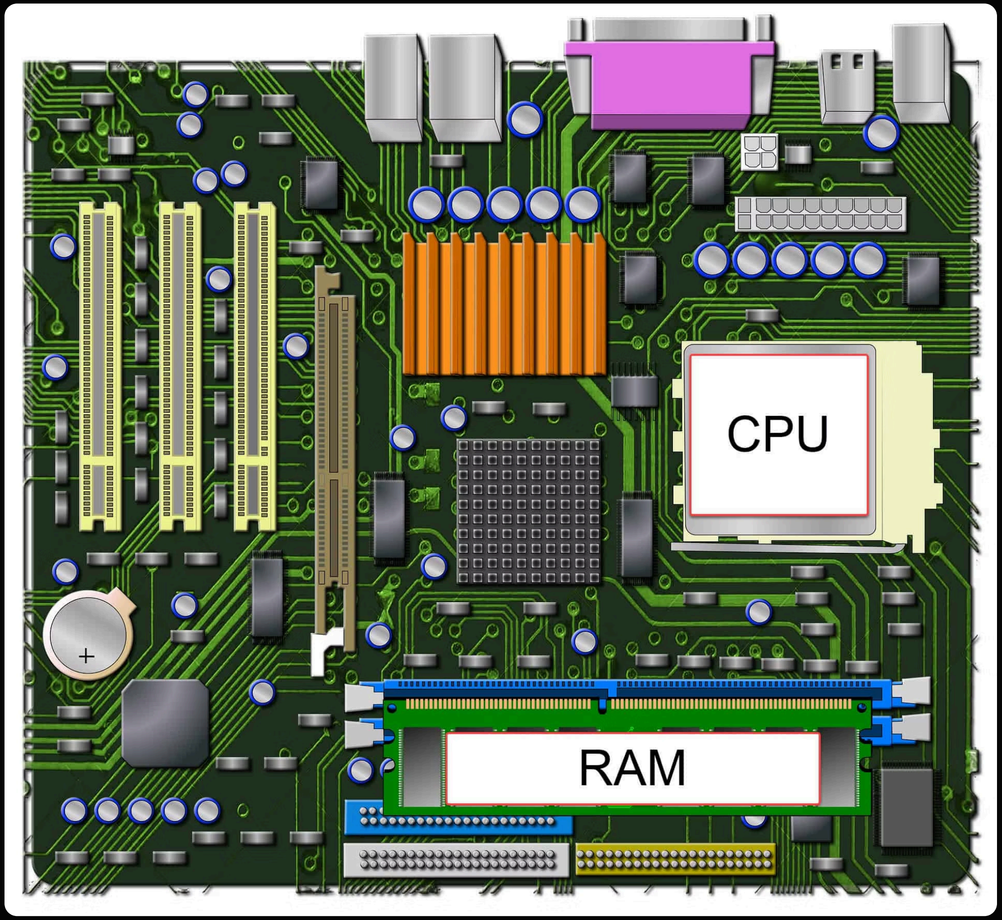
Hardware

Physical components you can touch and see. Performs tasks like storing data, processing information, and displaying results.

- CPU: Executes instructions
- Memory (RAM): Temporarily stores data
- Storage Devices: Permanently store data
- Input Devices: User interaction (keyboard, mouse)
- Output Devices: Display results (monitors, printers)

Software

Set of instructions telling the computer what, when, and how to do tasks. Examples: MS Paint, WhatsApp, web browsers, games.



Software converts input into machine-readable language for the CPU, which then produces output.

Core Components of a Computer

1

CPU (Central Processing Unit)

The "brain" of the computer. Executes instructions, performs calculations, and handles tasks. Comprises ALU (Arithmetic Logic Unit) and CU (Control Unit).

2

Motherboard

Main circuit board connecting and enabling communication between all computer components.

3

Memory (RAM)

Random Access Memory temporarily stores data for quick access while the computer is running.

4

Storage

Includes HDDs and SSDs that store data permanently, even when the computer is off.

5

Input/Output Devices

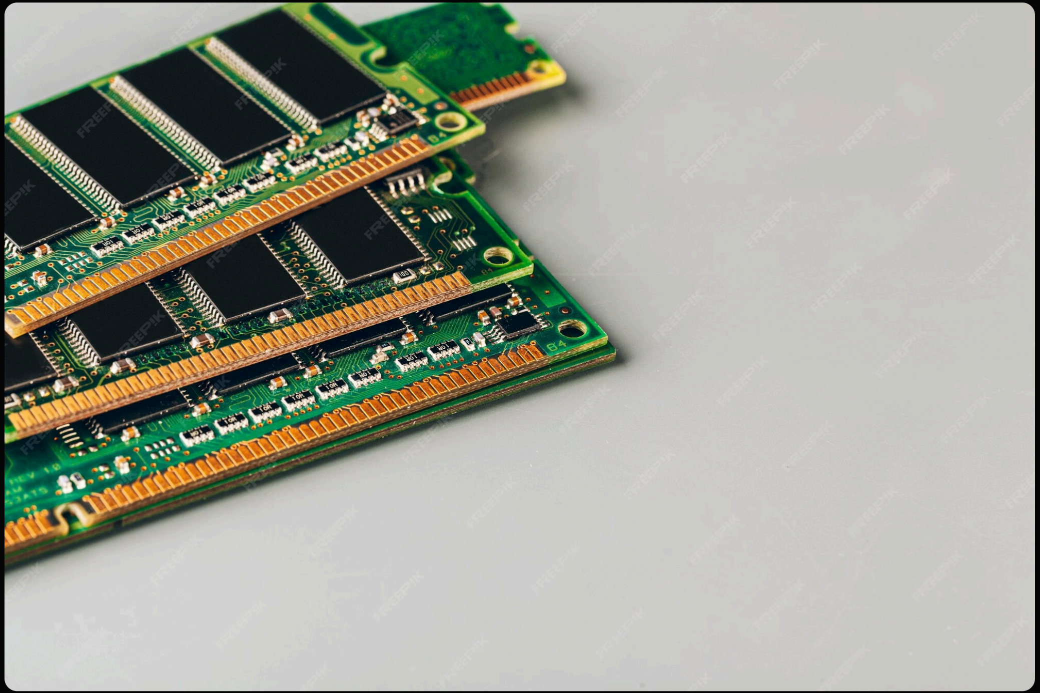
Input devices (keyboard, mouse) send data in; output devices (printer, speakers) display results.

Computer Memory: Primary vs. Secondary

Primary Memory (Main Memory)

Stores data and instructions during operations. Uses semiconductor technology.

- **RAM (Random Access Memory):** Volatile, temporary storage for active programs/data. Types: SRAM (faster), DRAM (slower, needs refreshing).
- **ROM (Read-Only Memory):** Non-volatile, stores boot-up instructions and firmware. Types: PROM, EPROM, EEPROM.



Secondary memory is non-volatile, stores large amounts of data permanently, and is slower than primary memory.

Generations of Computers

1

1st Gen (1940-1956)

Vacuum tubes, massive, machine language, punch cards. (ENIAC, UNIVAC)

2

2nd Gen (1956-1963)

Transistors, smaller, faster, introduced programming languages (FORTRAN, COBOL).

3

3rd Gen (1964-1971)

Integrated circuits (IC), semiconductors, smaller, cheaper, accessible to wider audience.

4

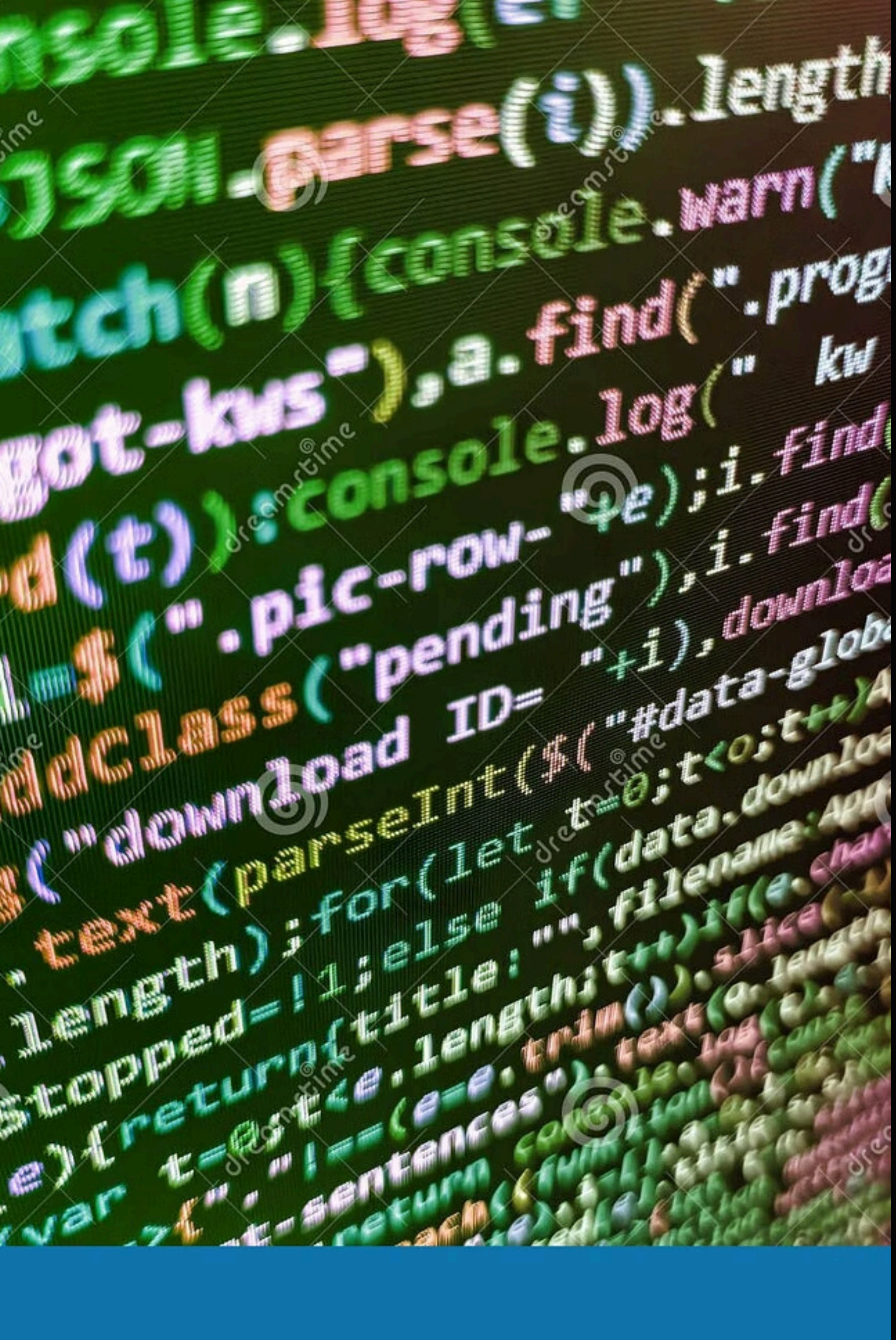
4th Gen (1971-Present)

Microprocessors, portable, user-friendly, personal computers, networks. (IBM PC, Apple II)

5

5th Gen (Present & Beyond)

AI-based technology, quantum computing, nanotechnology, parallel processing. (Smartphones, Desktops)



Programming Languages & Processors

Programming Languages

Set of instructions and syntax to create software. Key features include syntax, data types, variables, operators, control structures, libraries, and paradigms.

Examples: Python, Java, C++, JavaScript.

Language Processors

Convert source code into machine code. Find errors during translation.

- **Compiler:** Translates entire program at once (C, C++).
- **Assembler:** Translates assembly language into machine code.
- **Interpreter:** Translates and executes line by line (Python, Perl).

Algorithms: The Blueprint of Logic

What is an Algorithm?

A finite sequence of clear, unambiguous steps to solve a specific problem.

Why are Algorithms Needed?

Solve complex problems efficiently, automate processes, enable tasks impossible for humans.

Key Characteristics

Clear, well-defined inputs/outputs, finite, feasible, language-independent, effective.

Algorithms can be expressed in natural language, flowcharts, or pseudocode (a code-like structure in plain English).

Pseudocode: Translating Logic to Steps

Pseudocode serves as a bridge between human language and programming language. It offers a detailed, step-by-step description of an algorithm's logic using plain English and a code-like structure, making it understandable for both developers and non-technical stakeholders.

Organize Tasks & Set Goals

Clearly sequence tasks and state the main objective of the pseudocode at the outset.

Use Standard Structures

Employ common programming constructs like `IF-ELSE`, `FOR`, and `WHILE`. Indent for readability.

Naming & Keywords

Use simple, distinct naming conventions. Represent reserved keywords (e.g., IF, WHILE) in CAPITAL LETTERS.

Clarity & Simplicity

Ensure the pseudocode is finite, clear, and easy to understand for any audience, avoiding technical jargon.

Thank You!

Questions?