🔹 CPU (Central Processing Unit)

- The **"brain" of the computer.**
- Performs **calculations, decision-making, and instruction execution.**
- Has components like:
  - **ALU (Arithmetic Logic Unit)** → does math & logic.
  - **Control Unit (CU)** → directs operations.
  - **Registers** → very small, ultra-fast memory inside CPU.
- Very fast but has **small storage capacity.**

🔹 RAM (Random Access Memory)

- The **main memory** of the computer.
- Stores **data and instructions temporarily** while the CPU is working.
- Contents are **volatile** → data is lost when power goes off.
- Much **larger in size** than CPU registers, but **slower.**

⚡ Simple Analogy

- **CPU** = A person working (the thinker).
- **RAM** = The desk or workspace where the person keeps books & papers while working.
  - Bigger desk (more RAM) → you can keep more things at once.
  - Faster thinker (faster CPU) → you can process tasks quicker.

✅ In short:

- **CPU = Processing power (executes instructions).**
- **RAM = Working memory (stores data/instructions temporarily).**

Do you want me to also explain how **CPU, RAM, and Storage (Hard Disk/SSD)** are connected (step by step data flow)?

🔹 1. Storage (HDD/SSD)

- Permanent memory (non-volatile).
- Stores OS, apps, files, etc.
- Very **large in size**, but **slowest** among memory types.

🔹 2. RAM (Main Memory)

- Temporary memory, holds **programs and data currently in use.**
- Much **faster** than storage, but still slower than CPU.
- Volatile → data lost when power goes off.

🔹 3. Cache Memory

- A **tiny, ultra-fast memory** located **inside or very close to the CPU.**
- Stores **frequently used data/instructions** from RAM so CPU can access them quickly.
- Levels:
  - **L1 Cache** → Smallest & fastest, inside CPU core.
  - **L2 Cache** → Bigger, slightly slower.
  - **L3 Cache** → Largest, shared among cores, slower than L1/L2 but still faster than RAM.

🔹 4. CPU (Central Processing Unit)

- Executes instructions (calculations, logic, control).
- Needs data quickly, so it first checks:
  1. **Cache** → If data is there (Cache Hit) → super fast.
  2. If not in cache (Cache Miss) → fetch from **RAM.**
  3. If not in RAM → fetch from **Storage (HDD/SSD)** → slowest.

⚡ Data Flow Example

1. You open a game stored in **SSD/HDD.**
2. OS loads required files into **RAM.**
3. CPU fetches instructions/data from RAM.
4. Cache stores frequently accessed game data → CPU runs much faster without always waiting for RAM.