# **Introduction to Data Structures**

### **1. What is a Data Structure?**

**Definition**: A way of **organizing and storing data** so that it can be used efficiently.

Example: Think of a cupboard → Clothes arranged in racks (organized) → Easier to find.

### **2. Concept of ADT (Abstract Data Type)**

**Definition**: ADT defines **what operations** can be performed on data, but **not how** they are implemented.

Example:

**Stack ADT**: Operations → push, pop, peek.

How stack is built (array or linked list) = hidden.

### **3. Types of Data Structures**

#### **(A) Linear Data Structures**

👉 Data is arranged **sequentially** (one after another).

Examples:

**Array** – fixed size, stores elements in continuous memory.

**Linked List** – nodes connected using pointers.

**Stack** – LIFO (Last In, First Out).

**Queue** – FIFO (First In, First Out).

#### **(B) Non-Linear Data Structures**

👉 Data is arranged in **hierarchy or network** form.

Examples:

**Tree** – hierarchical structure (root → branches → leaves).

**Graph** – nodes (vertices) connected by edges (like a social network).

#### **(C) Static Data Structures**

👉 Size is **fixed** (cannot grow/shrink **at runtime**).

Example: Array.

#### **(D) Dynamic Data Structures**

👉 Size can **grow or shrink** at runtime.

Example: Linked List, Stack, Queue (when implemented dynamically).

### **4. Operations on Data Structures**

Most data structures support these **basic operations**:

**Traversal** – Visit each element.

**Insertion** – Add new data.

**Deletion** – Remove data.

**Searching** – Find data.

**Sorting** – Arrange data (ascending/descending).

**Updating** – Modify existing data.