9th Task in Embedded Systems

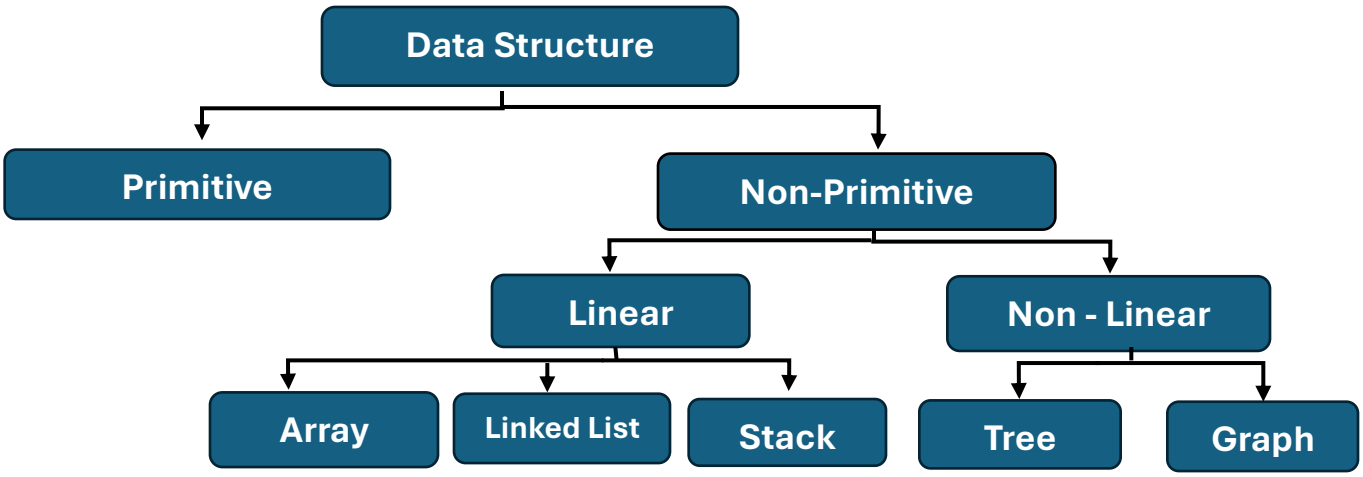
What is a Data Structure?

A **data structure** is a way to store and organize data so that it can be used efficiently.

Types of Data Structures:

1. **Primitive** (Basic) – int, float, char, boolean
2. **Non-Primitive** (Complex)
   - **Linear** → Elements arranged sequentially (Arrays, Linked Lists, Stacks, Queues)
   - **Non-Linear** → Elements not in sequence (Trees, Graphs)

Mind Map Sketch:



Overview of every kind of data structures?

Linear Data Structures

Elements are arranged sequentially.

-Examples:

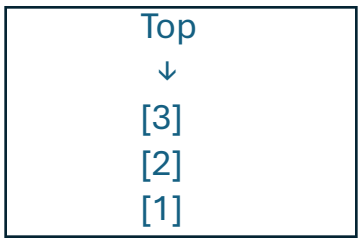
- **Array :** Fixed-size collection ([1, 2, 3]).

```
Index:  0   1   2   3
Value: [10, 20, 30, 40]
```

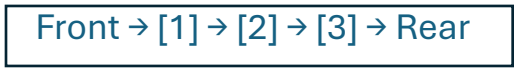
- **Linked List :** Nodes connected via pointers (1 → 2 → 3).

```
Head → [1 | →] → [2 | →] → [3 | →] → NULL
```

- **Stack :** LIFO (Last In, First Out) like a stack of books.



- **Queue :** FIFO (First In, First Out) like a ticket line.



Non-Linear Data Structures

- Elements are **not in a sequence**; hierarchical or networked.
- **Examples**:
  - **Tree** : Hierarchical.
  - **Graph** : Nodes + Edges .

◈ What is linked list and its types and its applications?

A **linked list** is a chain of nodes where each node has **data + pointer** to the next node.

Types:

- **Singly Linked List** : 1 → 2 → 3 → NULL (One-directional)
- **Doubly Linked List** : 1 ⇄ 2 ⇄ 3 (Two-way traversal)
- **Circular Linked List** : 1 → 2 → 3 → (back to 1) (Loop)

Applications:

- Used in **browser history, undo operations, dynamic memory allocation**.

◈ Stack and Queue and difference between them (Comparison)?

Feature	Stack	Queue
Insertion	`push()` at top	`enqueue()` at rear
Deletion	`pop()` from top	`dequeue()` from front
Example	Browser "Back" button	Printer job scheduling