**DATA STRUCTURE**

**Data Types:**

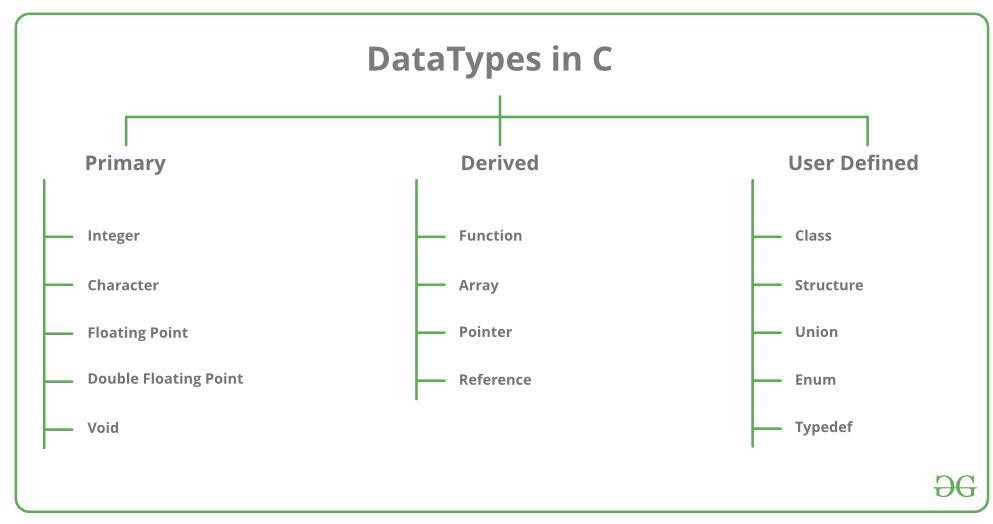
Data types are means to identify the type of data and associated operations of handling it.

There are three types of data types:

1. Pre-defined DataTypes

2. Derived Data Types

3. User-defined DataTypes



The data types in C can be classified as follows:

Primitive Data Types

1.Primitive data types are the most basic data types that are used for representing simple values such as integers, float, characters, etc.

2.User Defined Data Types The user-defined data types are defined by the user himself.

3.Derived Types The data types that are derived from the primitive or built-in datatypes are referred to as Derived Data Types

Q.What is Data Structure:

A data structure is a storage that is used to store and organize data.

A data structure is not only used for organizing the data.

So we must have good knowledge of data structures.

Data structures are an integral part of computers used for the arrangement of data in memory.

They are essential and responsible for organizing, processing, accessing, and storing data efficiently.

**Data Structure:**

The data type is the form of a variable to which a value can be assigned

It can hold value but not data. Therefore, it is dataless

The implementation of a data type is known as abstract implementation

There is no time complexity in the case of data types.

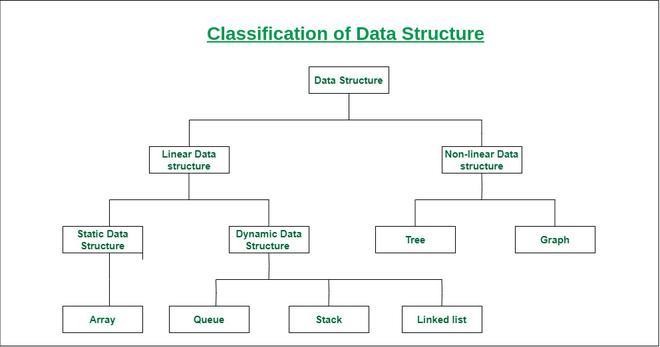
In the case of data types, the value of data is not stored

Data type examples are int, float, double, etc.

Need Of Data structure :

1. Data structure modification is easy. 2. It requires less time. 3. Save storage memory space. 4. Data representation is easy. 5. Easy access to the large database.

|  |  |
| --- | --- |
| **Data Type** | **Data Structure** |
| The implementation of a data type is known as abstract implementation. | Data structure implementation is known as concrete implementation. |
| There is no time complexity in the case of data types. | In data structure objects, time complexity plays an important role. |
| In the case of data types, the value of data is not stored because it only represents the type of data that can be stored. | While in the case of data structures, the data and its value acquire the space in the  computer’s main memory. Also, a data structure can hold different kinds and types of data within one single object. |
| Data type examples are int, float, double, etc. | Data structure examples are stack, queue, tree, etc. |

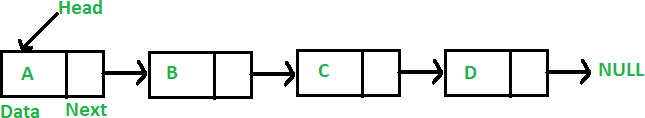


# Linked list:

A linked list is a linear data structure in which elements are not stored at contiguous memory locations. The elements in a linked list are linked using pointers as shown in the below image:

Types of linked lists:

* + Singly-linked list
  + Doubly linked list
  + Circular linked list
  + Doubly circular linked list



*Linked List*

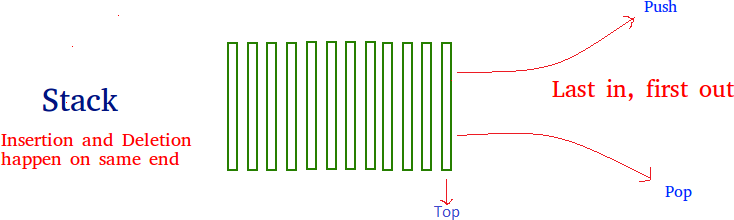
# Characteristics of a Linked list:

A linked list has various characteristics which are as follows:

* + A linked list uses extra memory to store links.
  + During the initialization of the linked list, there is no need to know the size of the elements.
  + Linked lists are used to implement stacks, queues, graphs, etc.
  + The first node of the linked list is called the Head.
  + The next pointer of the last node always points to NULL.
  + In a linked list, insertion and deletion are possible easily.
  + Each node of the linked list consists of a pointer/link which is the address of the next node.
  + Linked lists can shrink or grow at any point in time easily.

# Stack:

Stack is a linear data structure that follows a particular order in which the operations are performed. The order is [LIFO(Last in first](https://www.geeksforgeeks.org/lifo-last-in-first-out-approach-in-programming/) [out)](https://www.geeksforgeeks.org/lifo-last-in-first-out-approach-in-programming/). Entering and retrieving data is possible from only one end. The entering and retrieving of data is also called push and pop operation in a stack. There are different operations possible in a stack like reversing a stack using recursion, Sorting, Deleting the middle element of a stack, etc.



# Characteristics of a Stack:

Stack has various different characteristics which are as follows:

* + Stack is used in many different algorithms like Tower of Hanoi, tree traversal, recursion, etc.
  + Stack is implemented through an array or linked list.
  + It follows the Last In First Out operation i.e., an element that is inserted first will pop in last and vice versa.
  + The insertion and deletion are performed at one end i.e. from the top of the stack.

# Operation performed on stack ;

* + **Push**: Elements can be pushed onto the top of the stack, adding a new element to the top of the stack.
  + **Pop**: The top element can be removed from the stack by performing a pop operation, effectively removing the last element that was pushed onto the stack.
  + **Peek:** The top element can be inspected without removing it from the stack using a peek operation.
  + **IsEmpty**: A check can be made to determine if the stack is empty.

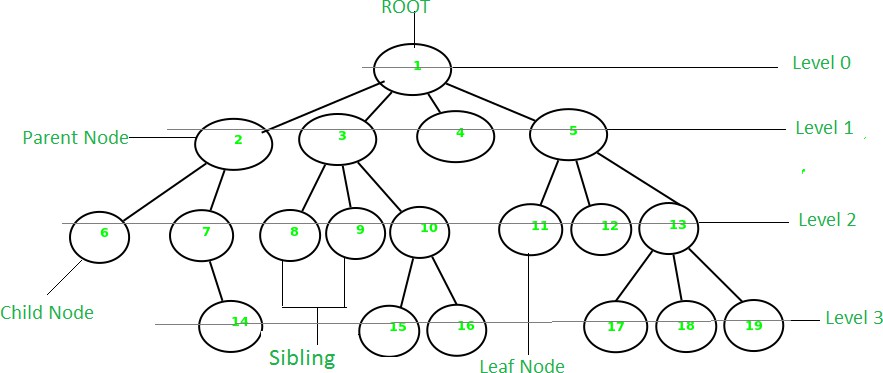
# Real-Life Applications of Stack:

# Tree:

A tree is a non-linear and hierarchical data structure where the elements are arranged in a tree-like structure. In a tree, the topmost node is called the root node. Each node contains some data, and data can be of any type. It consists of a central node, structural nodes, and sub-nodes which are connected via edges. Different tree data structures allow quicker and easier access to the data as it is a non- linear data structure. A tree has various terminologies like Node, Root, Edge, Height of a tree, Degree of a tree, etc.

There are different types of Tree-like

* + [Binary Tree](https://www.geeksforgeeks.org/binary-tree-data-structure/),
  + [Binary Search Tree](http://www.geeksforgeeks.org/binary-search-tree-set-1-search-and-insertion/),
  + [AVL Tree](https://www.geeksforgeeks.org/avl-tree-set-1-insertion/),
  + [B-Tree,](https://www.geeksforgeeks.org/introduction-of-b-tree-2/) etc.



*Tree*

# Characteristics of a Tree:

The tree has various different characteristics which are as follows:

* + A tree is also known as a Recursive data structure.
  + In a tree, the Height of the root can be defined as the longest path from the root node to the leaf node.
  + In a tree, one can also calculate the depth from the top to any node. The root node has a depth of 0.

# Applications of Tree:

Different applications of Tree are as follows:

* + Heap is a tree data structure that is implemented using arrays and used to implement priority queues.
  + B-Tree and B+ Tree are used to implement indexing in databases.
  + Syntax Tree helps in scanning, parsing, generation of code, and evaluation of arithmetic expressions in Compiler design.
  + K-D Tree is a space partitioning tree used to organize points in K-dimensional space.
  + Spanning trees are used in routers in computer networks.