**Data Structure**- Definition- A **data structure** is a particular way of organizing data in a computer so that it can be used effectively.

**Characteristics of data structures**

Data structures are often classified by their characteristics. Possible characteristics are:

Linear or non-linear: This characteristic describes whether the data items are arranged in chronological sequence, such as with an array, or in an unordered sequence, such as with a graph.

Homogeneous or non-homogeneous: This characteristic describes whether all data items in a given repository are of the same type or of various types.

Static or dynamic: This characteristic describes how the data structures are compiled. Static data structures have fixed sizes, structures and memory locations at compile time. Dynamic data structures have sizes, structures and memory locations that can shrink or expand depending on the use.

**Built-in & User Defined Data Types**



**Data Structures** are normally classified into two broad categories

1. Primitive Data Structure
2. Non-primitive data Structure

### Data types

A particular kind of data item, as defined by the values it can take, the programming language used, or the operations that can be performed on it.

**Primitive Data Structure**

* Primitive data structures are basic structures and are directly operated upon by machine instructions.
* Primitive data structures have different representations on different computers.
* Integers, floats, character and pointers are examples of primitive data structures.
* These data types are available in most programming languages as built in type.
  + Integer: It is a data type which allows all values without fraction part. We can use it for whole numbers.
  + Float: It is a data type which use for storing fractional numbers.
  + Character: It is a data type which is used for character values.

Pointer: A variable that holds memory address of another variable are called pointer.

### Non primitive Data Type

* These are more sophisticated data structures.
* These are derived from primitive data structures.
* The non-primitive data structures emphasize on structuring of a group of homogeneous or heterogeneous data items.
* Examples of Non-primitive data type are Array, List, and File etc.
* A Non-primitive data type is further divided into Linear and Non-Linear data structure
  + **Array:** An array is a fixed-size sequenced collection of elements of the same data type.
  + **List:** An ordered set containing variable number of elements is called as Lists.
  + **File:** A file is a collection of logically related information. It can be viewed as a large list of records consisting of various fields.

### Linear data structures

* A data structure is said to be Linear, if its elements are connected in linear fashion by means of logically or in sequence memory locations.
* There are two ways to represent a linear data structure in memory,
  + Static memory allocation
  + Dynamic memory allocation
* The possible operations on the linear data structure are: Traversal, Insertion, Deletion, Searching, Sorting and Merging.
* Examples of Linear Data Structure are Stack and Queue.
* Stack: Stack is a data structure in which insertion and deletion operations are performed at one end only.
  + The insertion operation is referred to as ‘PUSH’ and deletion operation is referred to as ‘POP’ operation.
  + Stack is also called as Last in First out (LIFO) data structure.
* Queue: The data structure which permits the insertion at one end and Deletion at another end, known as Queue.
  + End at which deletion is occurs is known as FRONT end and another end at which insertion occurs is known as REAR end.
  + Queue is also called as First in First out (FIFO) data structure.

### Nonlinear data structures

* Nonlinear data structures are those data structure in which data items are not arranged in a sequence.
* Examples of Non-linear Data Structure are Tree and Graph.
* **Tree:** A tree can be defined as finite set of data items (nodes) in which data items are arranged in branches and sub branches according to requirement.
  + Trees represent the hierarchical relationship between various elements.
  + Tree consist of nodes connected by edge, the node represented by circle and edge lives connecting to circle.
* **Graph:** Graph is a collection of nodes (Information) and connecting edges (Logical relation) between nodes.
  + A tree can be viewed as restricted graph.
  + Graphs have many types:
    - Un-directed Graph
    - Directed Graph
    - Mixed Graph
    - Multi Graph
    - Simple Graph
    - Null Graph
    - Weighted Graph

## Difference between Linear and Non Linear Data Structure

|  |  |
| --- | --- |
| **Linear Data Structure** | **Non-Linear Data Structure** |
| Every item is related to its previous and next time. | Every item is attached with many other items. |
| Data is arranged in linear sequence. | Data is not arranged in sequence. |
| Data items can be traversed in a single run. | Data cannot be traversed in a single run. |
| Eg. Array, Stacks, linked list, queue. | Eg. tree, graph. |
| Implementation is easy. | Implementation is difficult. |