

## 1. STACKS:

A stack is a linear data structure in which elements are added or removed from only one end of the stack, which is called 'top'. Stacks are based on the Last-in-First-out principle (LIFO) or First-in-Last-Out (FILO) principle since the last element added to the stack is the first one to be removed and vice-versa. Stack data structure is analogous to a stack of books or a stack of plates or a stack of CDs because one can add or remove a book, plate or a CD only at the top, and nowhere in between. The point of addition or deletion of elements (i.e., the top of the stack) is identified by a pointer. The main operations performed on a stack are-

1. Push (): Addition of an element to the stack
2. Pop (): Deletion (or reading) an element from the stack

A stack can be implemented using Arrays or Linked Lists. Stack implementation refers to choosing data storage mechanism and implementing the push and pop operations. Major uses of stacks are in string reversal (where the string can be reversed using the pop operation) and postfix and prefix conversion.

## 2. QUEUES:

Queue is also a linear data structure like stack, but differs from stacks as addition of elements occurs at a fixed end (we can call it the rear end) and deletion of elements occurs at a different fixed end (call it the front). Queues are based on the FIFO (first in first out) principle as the first element to enter the queue is the first element to be implemented (i.e., leave it). It is analogous to some real-life situations like a queue of people standing in a ticket counter, since a new person can join only at the end of the queue and a person who reaches the front can only leave. A queue of print requests from different computers attached to the same printer is also an example. Queues are also stored in contiguous memory locations (in case of array implementation). We need two pointers to specify the front and rear ends of the queue. The main operations performed on a queue are-

1. Insert (enqueue): Addition of an element to the rear (back end) the queue
2. Delete (dequeue): Deletion (or removing) an element from the front of the queue

Queues can also be implemented using Arrays or Linked Lists. Queues are important wherever data needs to be processed in order.

## 3. HASH-TABLES:

Hash-Table is a data structure that uses the Hash function to search elements in a quick time. This searching technique is called Hashing and is useful whenever the user has to deal with a large number of databases. The hash function is a mathematical function that accepts values called 'keys' as inputs and generates output values called 'hash-keys'. The elements corresponding to hash-keys are stored in the hash-table. In other words, the hash function contains the rule to generate a hash key that specifies the position in the hash table where the corresponding data will be stored. Sometimes, the hash function generates the same hash key for different key values i.e., the hash function is not one-one. This condition is called *collision*. Techniques such as linear probing are used to avoid collisions. Real life examples can be seen in school and college student records, where a student is identified by a unique id, which can be used to access the details about the students.