Data Structures

Introduction to Linked List,stack and Queue

ARRAY:

➢ Array is a linear data structure.

➢ The size of an array is constant and total size of the allocated memory is
   Equal to the upperlimit of the array index.

➢ Elements in an array are stored in contiguous locations.

➢ Single dimensional and Multi dimensional arrays.

➢ Elements of an array can be accessed by using indices,means we can access
   Any element of an array randomly.

➢ Static memory.

Linked list:

↓	↓
Datapart	pointer

- ➢ <u>What is a linked list??</u>

    A linked list is a linear collection of data elements which are actually called as nodes(node is a structure which contains any datatype and a pointer which always points to the next element) and it is a data structure in which group of nodes are linked in a sequence.

- ➢ Like array linked list is also a linear data structure.
- ➢ Unlike array size of linked list is not fixed and memory is allocated Dynamically.
- ➢ Elements of linked list are not stored contiguously as address of node is stored in previous node.

- ➢ Single,double and circular linked list.
- ➢ Elements of linked list can't be accessed randomly but can be accessed Sequentially.

Representation of a linked list:

➤ Each node is a structure which contains a data part and a pointer to the next node

➤ First node is called as head node and the last node pointer points to null.

```
Typedef struct linkedlist{
    Int data;
    struct linkedlist *next;
}linkedlist;
```

Types of linked list:

➤ **Singly linked list**

- only one pointer is maintained and only forward traversing is possible

```
Typedef struct linkedlist{
    Int data;
    struct linkedlist *next;
}linkedlist;
```

➤ **Doubly linked list**

- Two pointers are maintained,one will keep the address of previous node
  And the other keeps the address of the next node.In doubly linked list
  We can move in both forward and backward directions.

```
Typedef struct d_linkedlist{
    Int data;
    struct linkedlist *next;
    struct linkedlist *previous;
}d_linkedlist;
```

➢ **Circular linked list**

- Circular linked list may be a single or a doubly linked list but in circular
  Linked list we always point the last node to the head pointer(In normal
  Linked list we point the last node to null).

Disadvantages(drawbacks) of linked list:

➢ Linked list needs extra memory compared to arrays because we use pointer

For storing the address of next node.

➢ Random accessing of an element is not possible,for accessing any element
We have to traverse the entire linked list.

➢ Nodes are stored incontiguously which increases the time for accessing

Individual elements in the linked list.

➢ Reverse traversing becomes difficult incase of singly linked list but in case
of doubly linked list it requires another pointer(more memory).

STACK:

➢ A stack is a container of objects that are inserted and removed according To last-in first-out principle.
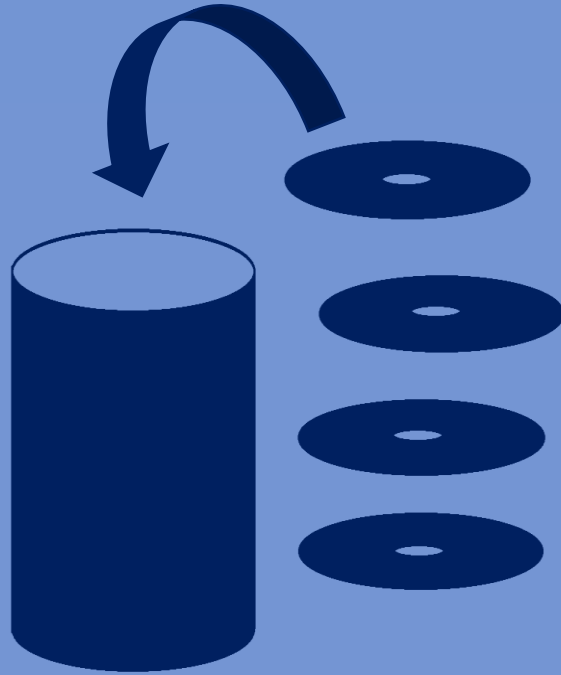
QUEUE:

➢ A queue is a container of objects that are inserted and removed according to the first-in first-out principle.