SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with ‘A’ Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING-IoT Including CS & BCT

COURSE NAME :23ITB201-DATA STRUCTURES & ALGORITHMS

II YEAR / III SEMESTER

Unit II- **STACK ADT & QUEUE ADT**

Topic : Stack ADT

Stack

A stack is a linear data structure where elements are stored in the LIFO (Last In First Out) principle where the last element inserted would be the first element to be deleted.

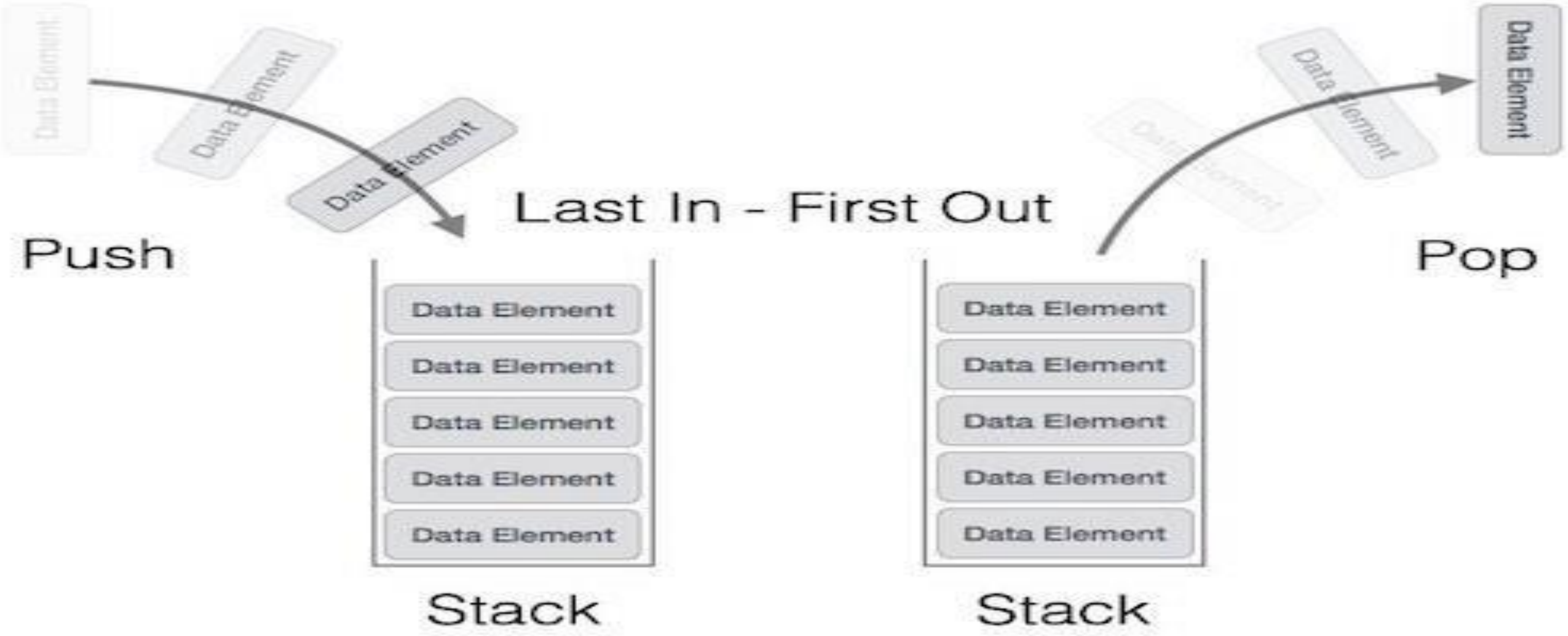
A stack is an Abstract Data Type (ADT), that is popularly used in most programming languages.

It is named stack because it has the similar operations as the real-world stacks, for example – a pack of cards or a pile of plates, etc.

Stack Representation

A stack allows all data operations at one end only.

At any given time, we can only access the top element of a stack.



A stack can be implemented by means of Array, Structure, Pointer, and Linked List.

Stack can either be a fixed size one or it may have a sense of dynamic resizing

Basic Operations on Stacks

The fundamental operations in the stack ADT include:

1. push()
2. pop()
3. peek()
4. isFull()
5. isEmpty().

These are all built-in operations to carry out data manipulation and to check the status of the stack.

Stack uses pointers that always point to the topmost element within the stack, hence called as the top pointer.

Stack Insertion: push()

The push() is an operation that inserts elements into the stack.

```
 int isfull(){
   if(top == MAXSIZE)
      return 1;
   else
      return 0;
}
```

```
/* Function to insert into the stack */
int push(int data){
   if(!isfull()) {
      top = top + 1;
      stack[top] = data;
   } else {
      printf("Could not insert data, Stack is full.\n");
   }
}
```

Stack Deletion: pop()

The pop() is a data manipulation operation which removes elements from the stack.

```
/* Check if the stack is empty */
int isempty()
{
  if(top == -1)
    return 1;
    else
    return 0;
}
```

```
/* Function to delete from the stack */
int pop()
{
  int data;
  if(!isempty()) {
    data = stack[top];
    top = top - 1;
    return data;
  } else {
    printf("Could not retrieve data, Stack is empty.\n");
  }
}
```

Retrieving topmost Element from Stack: peek()

The peek() is an operation retrieves the topmost element within the stack, without deleting it.

This operation is used to check the status of the stack with the help of the top pointer.

```
/* Function to return the topmost element in the stack */
int peek()
{
  return stack[top];
}
```

Any Query????

Thank you……