# STACK OPERATIONS

NAME – ADITYA KUMAR
ROLL NO – 34230821034
SUBJECT CODE – PCCCS-301

## TABLE OF CONTENTS

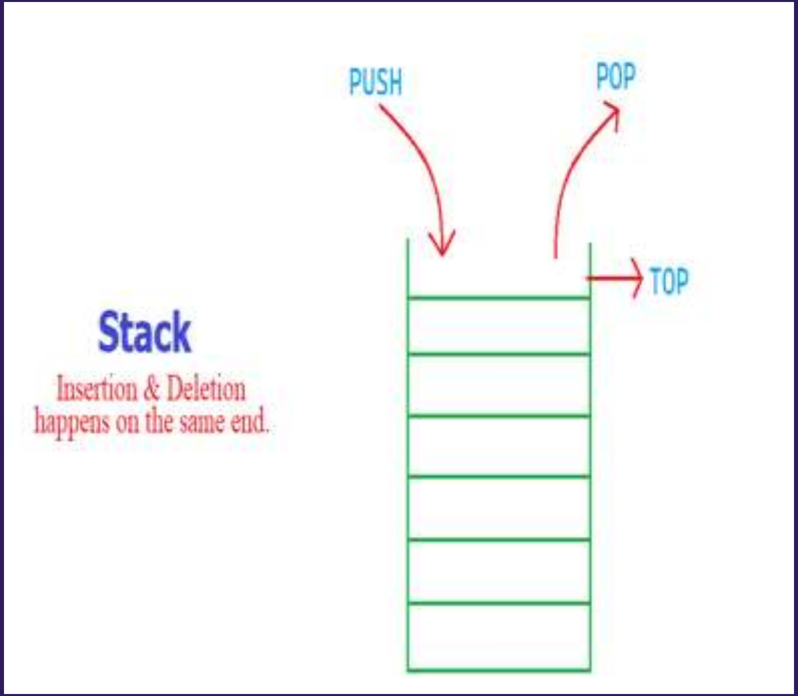
## What is Stack?

1. A stack is an Abstract Data Type (ADT), commonly used in most programming languages. It is named stack as it behaves like a real-world stack, for example – a deck of cards or a pile of plates, etc.
2. A real-world stack allows operations at one end only. For example, we can place or remove a card or plate from the top of the stack only. Likewise, Stack ADT allows all data operations at one end only. At any given time, we can only access the top element of a stack.
3. This feature makes it LIFO data structure. LIFO stands for Last-in-first-out.

## Operations of Stacks

The Stack data structure has two primary operations:

- ✓ **Push:** Adds an item to the stack. If the stack is full, then it is said to be an Overflow condition.
- ✓ **Pop:** Removes an item from the stack. The items are popped in the reversed order in which they are pushed. If the stack is empty, then it is said to be an Underflow condition.



## The PUSH( ) Operation

- Step 1: Checks is the stack is full.
- Step 2: If the stack is full, display “**overflow**” and exit.
- Step 3: If the stack isn’t full, increments **top** to point next empty space.
- Step 4: Ads data element to the stack location, where **top** is pointing.
- Step 5: **PUSH** Operation Completed Successfully.

- If the linked list is used to implement the stack, then in step 3, we need to allocate space dynamically.

## Algorithm of push Operation

A simple algorithm for Push operation can be derived as follows:

```
begin procedure push: stack, data
      if stack is full
         return null
      endif

      top ← top + 1
      stack[top] ← data
 end procedure
```

# Example of push with a C program

```
Void push(int value){
         if(top = size -1){
               Printf(“STACK is full”);
else
{
      top++;
       STACK[top] = value;
       Printf(“inserted element is %d”, STACK[top]);
      }
   }
}
```

## The POP( ) Operation

- Step 1: Checks is the stack is empty.
- Step 2: If the stack is empty, display “**underflow**” and exit.
- Step 3: If the stack isn’t empty, accesses the data element at which **top** is pointing..
- Step 4: Decreases the value of top by 1.
- Step 5: **POP** Operation Completed Successfully.

## Algorithm of POP Operation

A simple algorithm for Push operation can be derived as follows:

```
begin procedure pop: stack
        if stack is empty
            return null
        endif
        data ← stack[top]
        top ← top - 1
        return data
 end procedure
```

## Example of Pop( ) with C program

```
Void pop(){
      if(top==1){
            Printf("underflow condition is there");
else{
         Value=STACK[top];
         Top--;
         Printf("the deleted elements is %d", value);
        }
     }
}
```

## Application of Stack in Data Structures

- Expression Evaluation and Conversion
- Backtracking
- Function Call
- Parentheses Checking
- String Reversal
- Syntax Parsing
- Memory Management

THANK

YOU