**STACK IN DATA STRUCTURE AND OPERATION OF STACK**

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.



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.

**Basic features of Stack**

1. Stack is an ordered list of similar data type.

2. Stack is a LIFO structure. (Last in First out).

3. push() function is used to insert new elements into the Stack and pop() is used to delete an element from the stack. Both insertion and deletion are allowed at only one end of Stack called Top.

4. Stack is said to be in Overflow state when it is completely full and is said to be in Underflow state if it is completely empty.

**Stack Representation**

The following diagram depicts a stack and its operation**.** ****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. Here, we are going to implement stack using arrays, which makes it a fixed size stack implementation.

**Basic Operations**

Stack operations may involve initializing the stack, using it and then de-initializing it. Apart from these basic stuffs, a stack is used for the following two primary operations −

push() − Pushing (storing) an element on the stack.

pop() − Removing (accessing) an element from the stack.

When data is PUSHed onto stack.

To use a stack efficiently, we need to check the status of stack as well. For the same purpose, the following functionality is added to stacks −

peek() − get the top data element of the stack, without removing it.

isFull() − check if stack is full.

isEmpty() − check if stack is empty.

At all times, we maintain a pointer to the last PUSHed data on the stack. As this pointer always represents the top of the stack, hence named top. The top pointer provides top value of the stack without actually removing it.

**Push operation on stack**

The process of putting a new data element onto stack is known as a Push Operation. Push operation involves a series of steps −

**Step 1 −** Checks if the stack is full.

**Step 2 −** If the stack is full, produces an error and exit.

**Step 3** − If the stack is not full, increments top to point next empty space.

**Step 4 −** Adds data element to the stack location, where top is pointing.

**Step 5 −** Returns success.

****