What is Stack?

A STACK is an ordered list where all operations are restricted at one end of the list known as the top. A STACK is container of objects that are inserted and removed according to the **last-in first-out** (***LIFO***) principle.

Application of the Stack

- ✓ Infix to Postfix /Prefix conversion
- ✓ "Undo" mechanism in text editors; this operation is accomplished by keeping all text changes in a stack.
- ✓ To **reverse a word**. You push a given word to stack – letter by letter – and then pop letters from the stack.
- ✓ Forward and backward features in web browsers
- ✓ An **"undo"** mechanism in text editors; this operation is accomplished by keeping all text changes in a stack.
- ✓ Undo/Redo stacks in Excel or Word.

.

Stack Operations

Stack operations are restricted to the elements at the top of the stack.

- ✓ **Push** – Push operation refers to inserting an element in the stack. Since there's only one position at which the new element can be inserted—Top of the stack, the new element is inserted at the top of the stack.

- ✓ **Pop**– pop operation refers to the removal of an element. Again, since we only have access to the element at the top of the stack, there's only one element that we can remove. We just remove the top of the stack.
- ✓ **Peek** – Peek operation allows the user to see the element on the top of the stack. The stack is not modified in any manner in this operation.

- ✓ **Is Empty**– *Check if stack is empty or not.* To prevent performing operations on an empty stack, the programmer is required to internally maintain the size of the stack which will be updated during push and pop operations accordingly. isEmpty() conventionally returns a boolean value: True if size is 0, else False.

Pushing is the function were placing an item on the top of the stack.

Popping is the removal of the item from the top. This type of behavior in the list processing is called **LIFO** (**Last–In–First–Out**).