Riphah International University
I-14 Main Campus
Faculty of Computing

Class:	Fall-2024	**Subject:**	Data Structures & Algorithms
Course Code:	CS 2124	**Lab Instructor:**	Zeeshan Ali

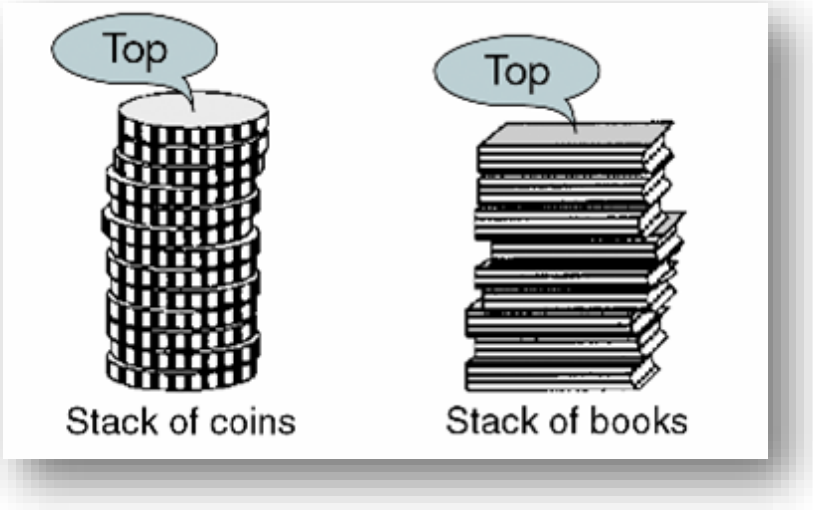
Learning Objective:

1. What is Stack?
2. Stack in C++.
3. Working of Stack?
4. Basic Operations of Stack?
5. Use of Stack?
6. Implementation of Stack using Array in C++.

7. Lab Tasks

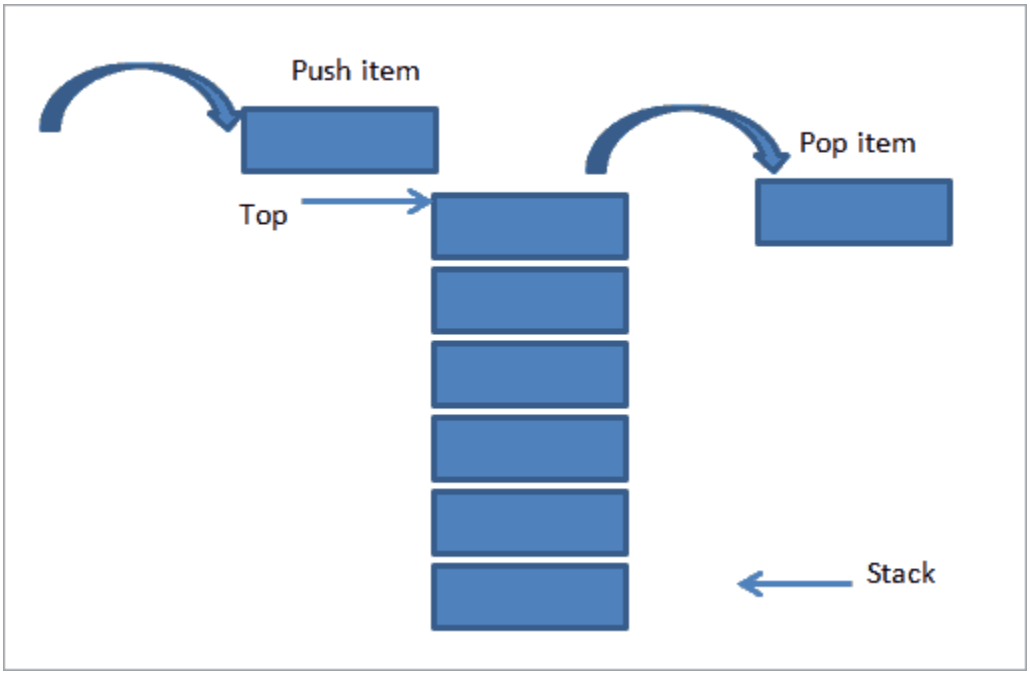
1. Stack:

Stack is a fundamental data structure which is used to store elements in a linear fashion. Stack follows FILO (First In Last Out) order or approach in which the operations are performed. This means that the element which was added last to the stack will be the first element to be removed from the stack.



Stack in C++

A stack is similar to real-life stack or a pile of things that we stack one above the other.



Working of Stack

As shown above, there is a pile of plates stacked on top of each other. If we want to add another item to it, then we add it at the top of the stack. This operation of adding an item to stack is called "**Push**".

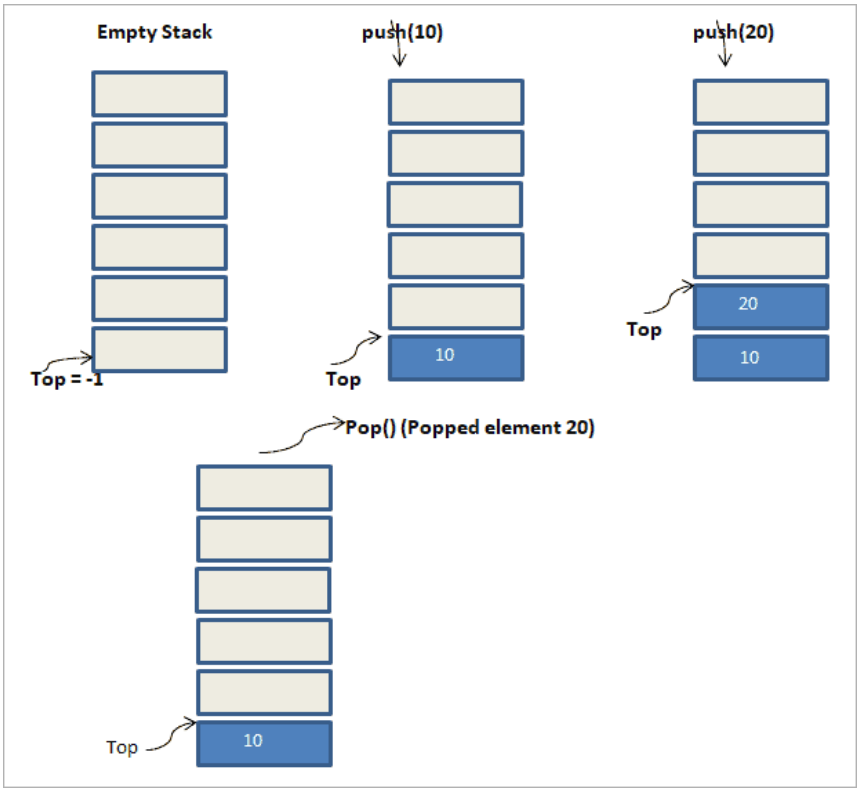
We remove an item from the stack. This is also done from the same end i.e. the top of the stack. This operation is called "**Pop**".

We see that push and pop are carried out from the same end. This makes the stack to follow FILO (First In Last Out) order. The position or end from which the items are pushed in or popped out to/from the stack is called the "**Top of the stack**".

Initially, when there are no items in the stack, the top of the stack is set to -1. When we add an item to the stack, the top of the stack is incremented by 1 indicating that the item is added. As opposed to this, the top of the stack is decremented by 1 when an item is popped out of the stack.

Basic Operations of Stack

- **push** – Adds or pushes an element into the stack.
- **pop** – Removes or pops an element out of the stack.
- **peek** – Gets the top element of the stack but doesn't remove it.
- **display** – Display all elements of the stack.



For an empty stack, the top of the stack is set to -1.

push:

- Next, we push the element 10 into the stack. We see that the top of the stack now points to element 10.
- We perform another push operation with element 20, as a result of which the top of the stack now points to 20

pop:

- We perform a pop () operation. As a result of the pop operation, the element pointed at the top of the stack is removed from the stack. We see that element 20 is removed from the stack. Thus the top of the stack now points to 10.

In this way, we can easily make out the FILO (First In Last Out) or LIFO approach used by stack.

Use of Stack

- Reverse a string
- Undo Mechanisms (i.e MS Word) etc.

Lab Task

- Implement Following Operations

  1)
  Stack (int ignored = 0)
  Requirements: None
  Results: Constructor. Creates an empty stack.
  2)
  ~Stack ()
  Requirements: None
  Results: Destructor. Deallocates (frees) the memory used to store a stack.
  3)
  void push (const DataItem)
  Requirements: None
  Results: Push the element at top of the stack.
  4)
  Void pop ()
  Requirements: Stack is not empty Results: Returns the
  element from the top of the stack.
  5)
  element Peek ()
  return element at the top of stack
  6)
  void clear ()
  Requirements: None
  Results: Removes all the elements from a stack.

  7)
  Bool isEmpty ()
  Requirements: None
  Results: Returns true if a stack is empty. Otherwise, returns false.

- Write a program in C++ to reverse a string (Data Structures) using stack.