

# Stack: Linear Data Structure

# Stack

- A Stack is a linear data structure that follows the LIFO (Last-In-First-Out) principle.
- A stack is an ordered list in which insertion and deletion are done at one end, called a top. The last element inserted is the first one to be deleted. Hence, it is called the Last in First out (LIFO) or First in Last out (FILO) list.

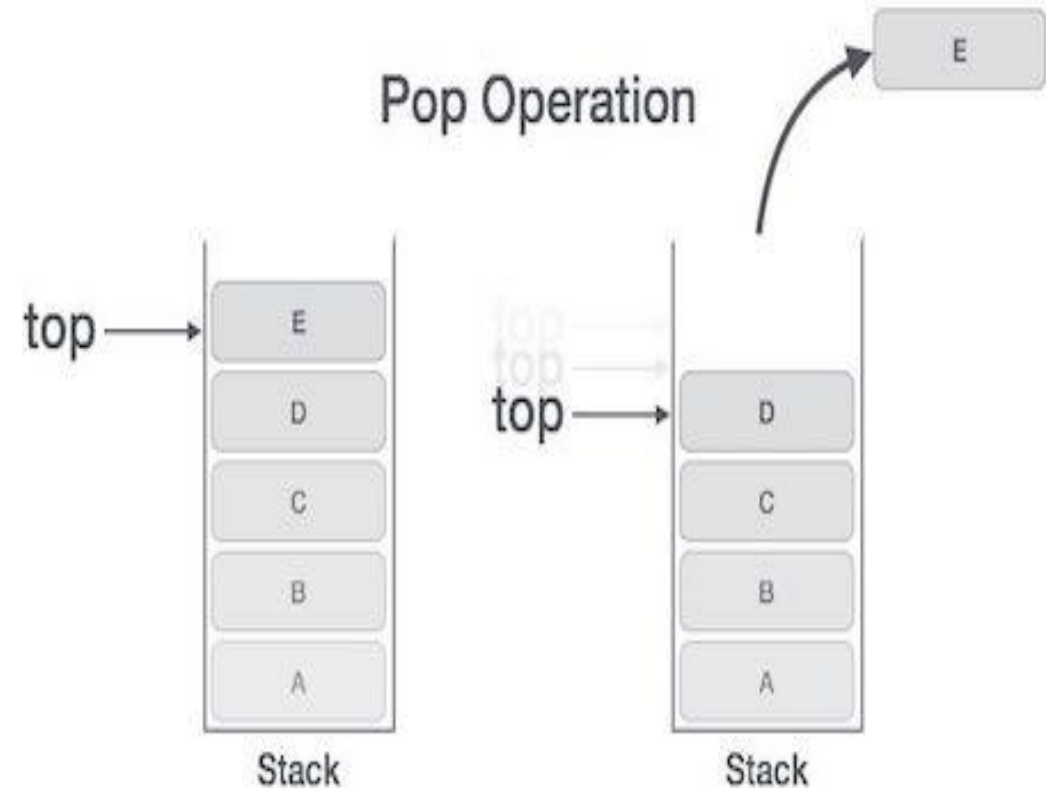
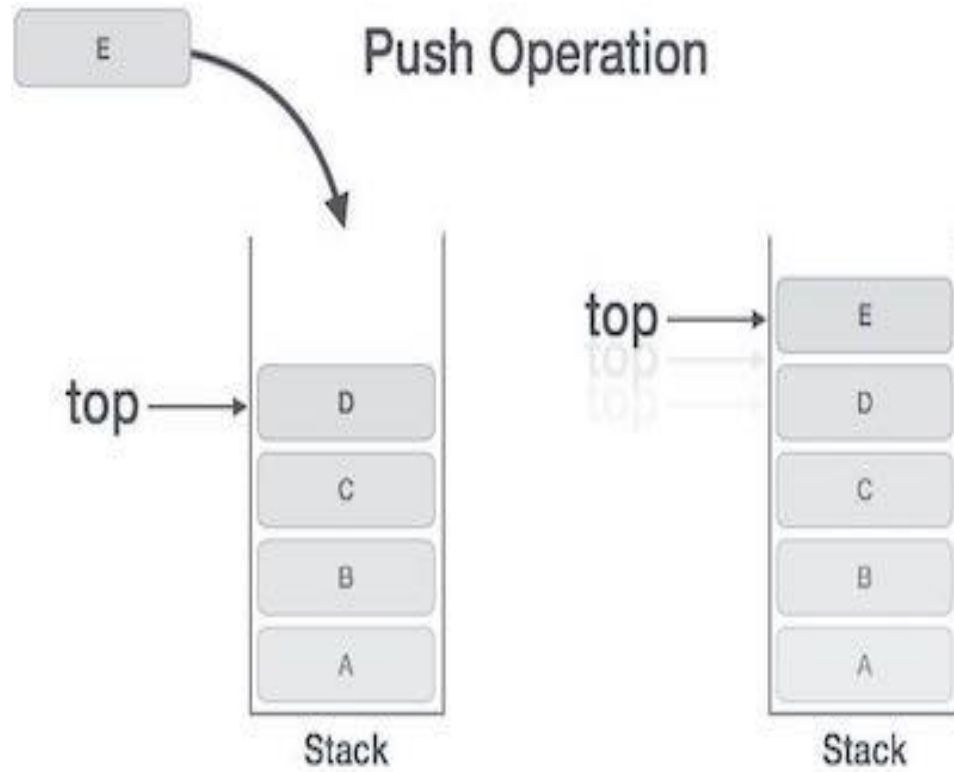
Stacks are a fundamental data structure. They are used in many applications.

- Example 1: Internet Web browsers store the addresses of recently visited sites on a stack. Each time a user visits a new site, that site's address is "pushed" onto the stack of addresses. The browser then allows the user to "pop" back to previously visited sites using the "back" button.
- Example 2: Text editors usually provide an "undo" mechanism that cancels recent editing operations and reverts to former states of a document. This undo operation can be accomplished by keeping text changes in a stack.

# Operations or functions of the Stack ADT

- `push(e)`: Adds element `e` to the top of the stack.
- `pop( )`: Removes and returns the top element from the stack (or null if the stack is empty).
- `top( )`: Returns the top element of the stack, without removing it (or null if the stack is empty).
- `size( )`: Returns the number of elements in the stack.
- `isEmpty( )`: Returns a boolean indicating whether the stack is empty.

# Stack Operations



Method	Return Value	Stack Contents
push(5)	—	(5)
push(3)	—	(5, 3)
size()	2	(5, 3)
pop()	3	(5)
isEmpty()	false	(5)
pop()	5	()
isEmpty()	true	()
pop()	null	()
push(7)	—	(7)
push(9)	—	(7, 9)
top()	9	(7, 9)
push(4)	—	(7, 9, 4)
size()	3	(7, 9, 4)
pop()	4	(7, 9)
push(6)	—	(7, 9, 6)
push(8)	—	(7, 9, 6, 8)
pop()	8	(7, 9, 6)

# Problem w.r.t stack operations

1. What values are returned during the following series of stack operations, if executed upon an initially empty stack? push(5), push(3), pop(), push(2), push(8), pop(), pop(), push(9), push(1), pop(), push(7), push(6), pop(), pop(), push(4), pop(), pop().
2. Suppose an initially empty stack S has performed a total of 25 push operations, 12 top operations, and 10 pop operations, 3 of which returned null to indicate an empty stack. What is the current size of S?

# Stack operations Implementation using Arrays

- Push() operation

## **Algorithm:**

begin

**if** top = n then stack full

        top = top + 1

        stack (top) := item;

end

## **Push Operation**

- In a push operation, an element into the top of the stack.
- Increment the variable Top so that it can now refer to the next memory location.
- Add an element at the position of the incremented top. This is referred to as adding a new element at the top of the stack.
- Throw an exception if Stack is full.

# Pop operation

## Pop()

### Algorithm :

begin

**if** top = 0 then stack empty;

    item := stack(top);

    top = top - 1;

end;

### Pop Operation

- Remove the top element from the stack and decrease the size of a top by 1.
- Throw an exception if Stack is empty.



## **Algorithm :**

PEEK (STACK, TOP)

Begin

**if** top = -1 then stack empty

item = stack[top]

**return** item

End

# Applications of Stack

- Balancing of symbols
- Infix-to-postfix conversion
- Evaluation of postfix expression
- Implementing function calls (including recursion)
- Page-visited history in a Web browser [Back Buttons]
- Undo sequence in a text editor
- Matching Tags in HTML and XML