

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.

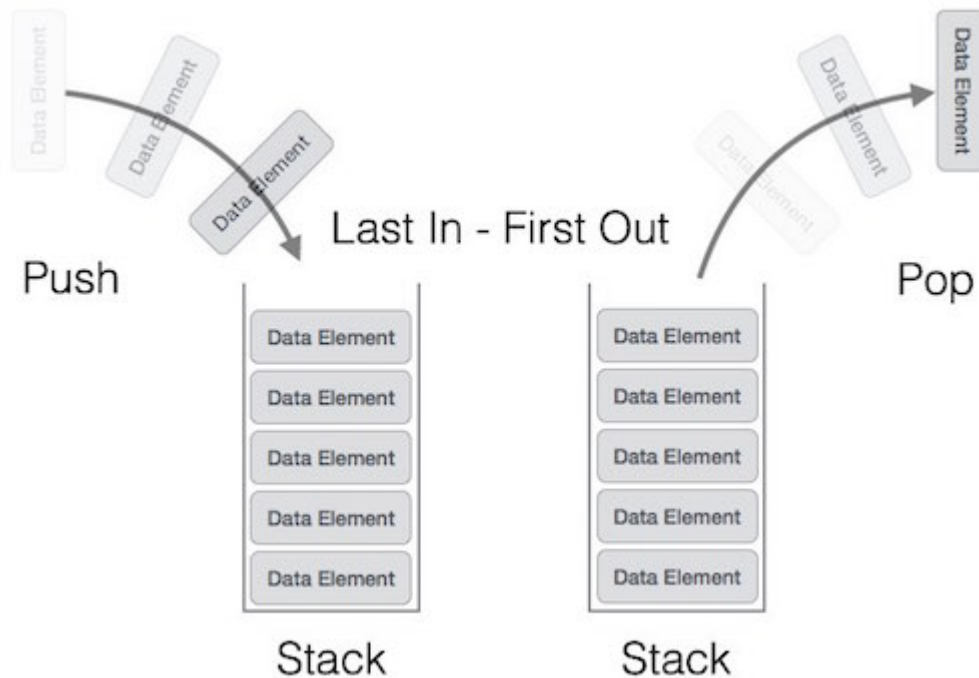


A real-world stack allows operations at one end only. For example, we can place or remove a card or plate from the top of the stack only. Likewise, Stack ADT allows all data operations at one end only. At any given time, we can only access the top element of a stack.

This feature makes it LIFO data structure. LIFO stands for Last-in-first-out. Here, the element which is placed (inserted or added) last, is accessed first. In stack terminology, insertion operation is called **PUSH** operation and removal operation is called **POP** operation.

Stack Representation

The following diagram depicts a stack and its operations –



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.

Application of Stack in Data Structure

Apart from the real-life example of stack here are some different applications of Stack in Data Structure. The functionality will be the same else we can't say it stack.

1. Memory management
2. Function Call(recursive functions.)
3. String Reversal
4. Parenthesis Checking
5. **Backtracking**
6. Syntax Parsing
7. Reversing a String
8. Matching HTML Tags in Web Developing.
9. Arithmetic Expression Evaluation
10. Java compiler uses postfix notation
11. Java virtual machine uses a stack
12. Expression Conversion or Expression Evaluation
 1. infix to prefix
 2. infix to postfix

Questions: The five items P,Q,R,S and T are pushed in a stack, one after the other starting from P. The stack is popped four times and each element is inserted in a queue. Then two elements are deleted from the queue and pushed back on the stack. now one item is popped from the stack. The popped item is:

Previous Year Questions

1. **What is the postfix expression of the string, $a+(b-c)*d$** [SBL-SO(IT/ICT)-2020] Ans. a
a) $abc-d*+$ b) $abcd*+$ c) $ad*bc-$ d) $abcd+*$ Ans:a

2. **Which data structure allow deleting data elements front and inserting at rear?**[Assistant programmer (ICB)- 2017] Ans. b
a) Stack b) Queue c) Dequeue d) Binary search tree

3. **The term push and pop related to** [Combined (AME)-2018] Ans. c
a) Array b) list c) stack d) all of this

4. **Which of the following name does not related to stacks?**[AP(ICB)- 2017, AP(HBFC, KB)-2018] Ans. a
a) FIFO List b) LIFO List c)Piles d) Push Down List
Explanation: FIFO is related to Queue.

5. **The term push and pop related to** [Combined (AME)-2018] Ans. c
a)Array b) list c) stack d) all of this

6. **Which of these data types is used by operating system to manage the Recursion in Java?**[Combined(IT/ICT-2018)] Ans. b
a)Array b) Stack c) Queue d) Tree

7. **Pushing an element into stack already having five elements and stack size of 5. Result in** [Combined (AP)-2018] Ans. a
a) Overflow b) Crash c) Underflow d) User flow

8. **The data structure required to check whether an expression contains balanced parenthesis is**[Combined (AP)-2018]Or Ans. a
The data structure required to evaluate a postfix expression is? [AP

(ICB)- 2017]

- a) Stack b) Queue c) Array d) Tree

9. Find the correct arranged data after stack operation? [BB (AP)-2016] Ans. a

push(1), push(2), pop, push(1), push(2), pop, pop, pop, push(2)

- (a) 2 2 1 1 2 (b) 2 2 1 2 1 (c) 2 2 2 2 1 (d) 2 2 2 1 2

Operation	Stack	Pop Sequence
Push 1	1	
Push 2	1, 2	
Pop	1	2
Push 1	1, 1	
Push 2	1, 1, 2	
Pop	1, 1	2, 2
Pop	1	2, 2, 1
Pop	Empty	2, 2, 1, 1
Push 2	2	
Pop	Empty	2, 2, 1, 1, 2

10. Stack operations are [AP (SBL)-2016] Ans. c

- a) delete, insertion b) insertion, delete
c) push, pop d) pop, raer

11. What will be the state of stack after executing following operation? Ans.: d

push(1), push(2), pop(), push(4), push(5), pop() [SBL&JBL(AME)-2020]

- a) 2, 5 b) 2, 4 c) 4, 5 d) 1, 4 Ans:d

Some Practices Question

1. Process of inserting an element in stack is called _____ Ans. b

- a) Create b) Push c) Evaluation d) Pop

2. Which one of the following is an application of Stack Data Structure? Ans. d

- a) Managing function calls b) The stock span problem
c) Arithmetic expression evaluation d) All of the above

3. What happens when you push a new node onto a stack? Ans. a

- a) The new node is placed at the front of the linked list
b) The new node is placed at the back of the linked list
c) The new node is placed at the middle of the linked list

d) No Changes happens

4. **In a stack, if a user tries to remove an element from empty stack it is called---** Ans. a

a) Underflow b) Empty collection c) Overflow d) Garbage Collection

5. **Entries in a stack are “ordered”. What is the meaning of this statement?** Ans. d

- a) A collection of stacks is sortable
- b) Stack entries may be compared with the '<' operation
- c) The entries are stored in a linked list
- d) There is a Sequential entry that is one by one

6. **The elements are removal from a stack in order.** Ans. a

- a) Reverse b) Hierarchical
- c) Alternative d) None

7. **Which of the following applications may use a stack?** Ans. d

- a) A parentheses balancing program
- b) Tracking of local variables at run time
- c) Compiler Syntax Analyzer
- d) All of the mentioned

8. **What is the value of the postfix expression 6 3 2 4 + - *** Ans. d

- a) Something between -5 and -15
- b) Something between 5 and -5
- c) Something between 5 and 15
- d) Something between 15 and 100

Explanation: On solving the postfix expression the answer comes out to 18

9. **Here is an infix expression: $4 + 3*(6*3-12)$. Suppose that we are using the usual stack algorithm to convert the expression from infix to postfix notation. The maximum number of symbols that will appear on the stack** Ans. d

