

# DSA through C++

## Stack



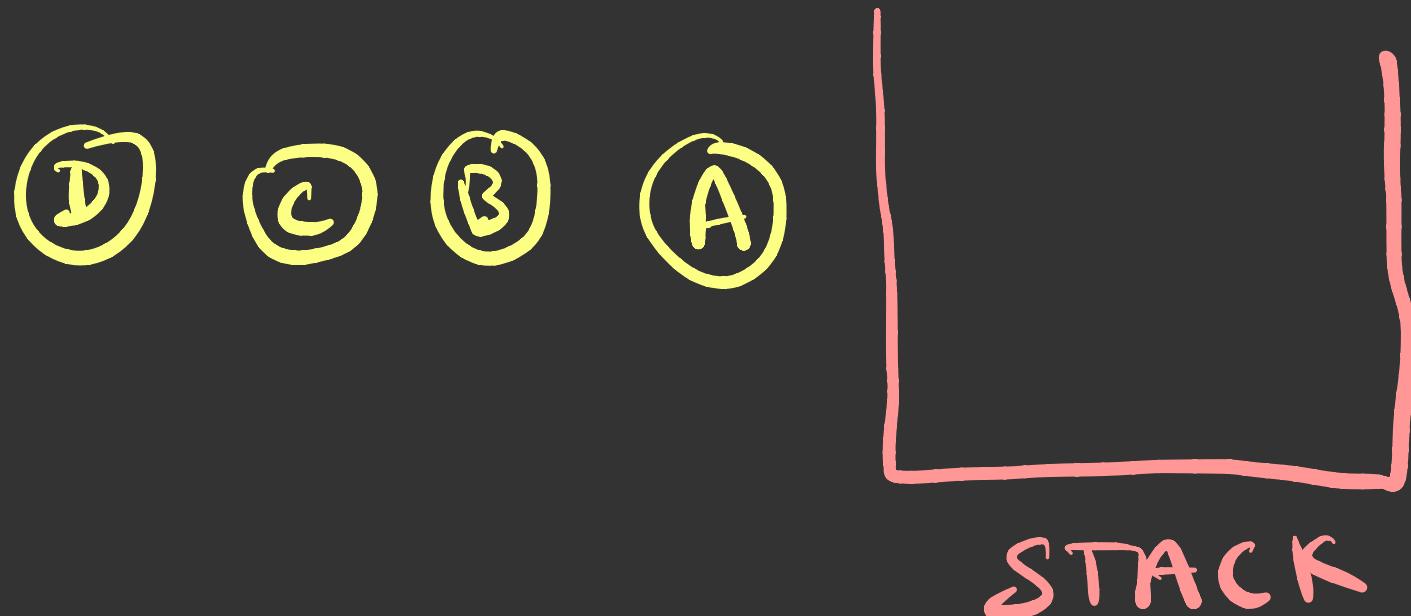
Saurabh Shukla (MySirG)

## Agenda

- ① what is STACK?
- ② Operations on STACK
- ③ Ways to implement STACK
- ④ Polish Notation
- ⑤ Algorithm to convert infix to postfix
- ⑥ Algorithm to evaluate postfix expression

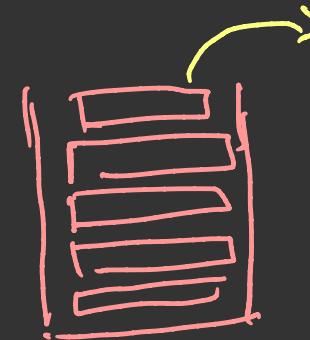
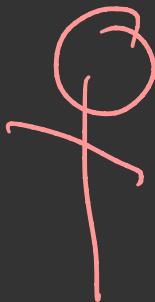
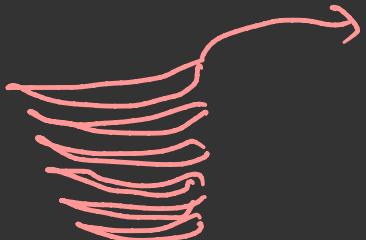
## What is STACK?

- STACK is a linear Data Structures
- STACK's working principle is Last In First Out (LIFO)



Travel

Real world examples of stack



Programming world examples of stack

Recursion | function call

$f_4()$

$f_3()$   
 $f_2()$   
 $f_1()$

# Operations on STACK

Insert

PUSH( )

Delete

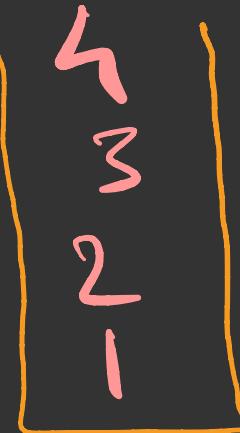
POP( )

View top  
element

PEEK( )

top

s



## Ways to implement STACK

- ① using Arrays
- ② using Dyn Arrays
- ③ using Linked List

## Polish Notation

The method of writing operators of an expression either before their operands or after them is called the Polish Notation

- Infix Notation
- Prefix Notation
- Postfix Notation

A + B

+ A B

A B +

## Practice

Infix :  $A + B * C$

Prefix :  $+ A * BC$

Postfix :  $ABC * +$

Infix :  $A * B - C / (D + E)$

Prefix :  $- * AB / C + DE$

Postfix :  $AB * CDE + / -$

Infix :  $A + B * (C - D) + E$

Prefix :  $+ + A * B - CDE$

Postfix :  $ABCD -* + E +$

$A * B - C / + DE$

$* AB - C / + DE$

$* AB - / C + DE$

$- * AB / C + DE$

$A * B - C / \underline{DE} +$

$AB * - C / DE +$

$AB * - CDE + /$

$AB * CDE + / -$



## Infix to Postfix

Suppose Q is an arithmetic expression written in infix notation. This algorithm finds the equivalent postfix expression P.

1. PUSH '(' onto the STACK and add ')' to the end of Q.
2. Scan Q from left to right and repeat steps 3 to 6 for each element of Q until the STACK is empty.
3. If an operand is encountered add it to P
4. If a left parenthesis is encountered, PUSH it onto the STACK.

5. If an operator (say #) is encountered, then:
- Repeatedly pop from STACK and add to P each operator which has the same precedence than #
  - Add # to STACK
6. If a right parenthesis is encountered, then:
- Repeatedly pop from the STACK and add to P each operator until a left parenthesis is encountered
  - Remove the left parenthesis
7. Exit.

## Evaluate Postfix

Let  $P$  be an arithmetic expression written in postfix notation. We uses a STACK to hold operands.

This algorithm finds the VALUE of an arithmetic expression  $P$  written in Postfix notation.

1. Add a right parenthesis ')' at the end of  $P$

2. Scan P from left to right and repeat step 3 and 4 for each element of P until the sentinel ')' is encountered.

3. If an operand is encountered, put it on the STACK

4. If an operator # is encountered, then  
a. Remove the two top elements of STACK, where X is the to element and Y is the next to top element

b. Evaluate Y # X

c. Place the result of (b) back on STACK

5. Set VALUE equal to the top element on STACK

6. Exit