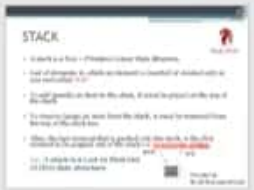
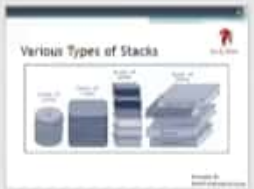




1

Study Point

Data Structures & Algorithms
Stack & its Applications

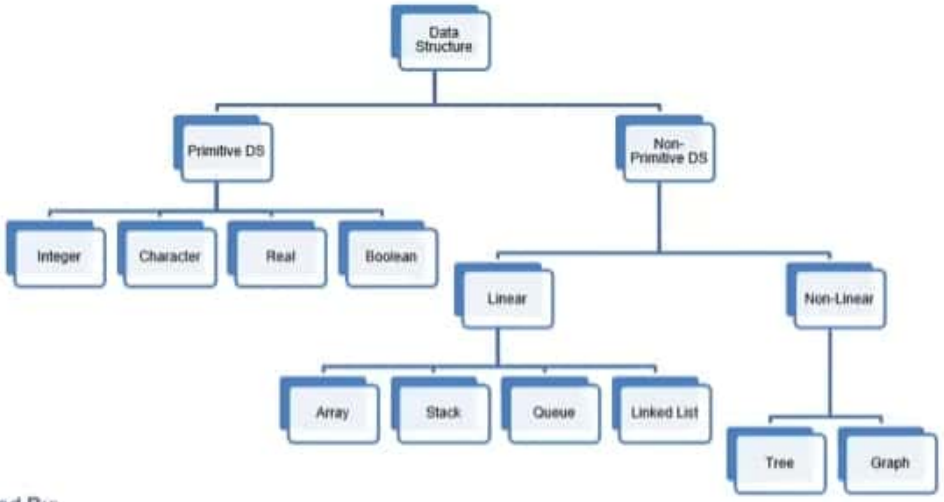
Presented By:

Sh. Muhammad Aamir
Lecturer
Department of Computer Science
GC University, Faisalabad

Click to add notes

Classification of Data Structures

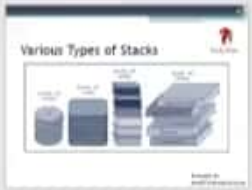
Study Point



Presented By:
Sheikh Muhammad Aamir

1 Data Structures & Algorithms Stack & its Applications
2 Classification of Data Structures
3 STACK
4 Various Types of Stacks
5 LIFO

Click to add notes

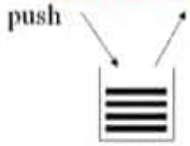


STACK

Study Point

- A stack is a Non – Primitive Linear Data Structure
- List of elements in which an element is inserted or deleted only at one end called TOP
- To add (**push**) an item to the stack, it must be placed on the top of the stack.
- To remove (**pop**) an item from the stack, it must be removed from the top of the stack too.
- Thus, the last element that is pushed into the stack, is the first element to be popped out of the stack i.e. **in reverse order**.

i.e., **A stack is a Last In First Out (LIFO) data structure**

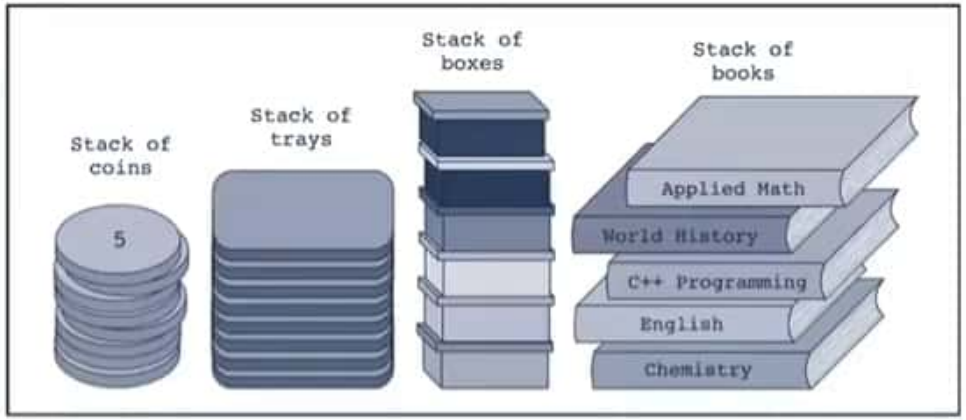


Presented By:
Sheikh Muhammad Aamir

Click to add notes

Various Types of Stacks

Study Point



Presented By:
Sheikh Muhammad Aamir

LIFO

Study Point

- Last In First Out (LIFO) data structure
  - Top element of stack is first element to be removed from stack
  - Elements added and removed from one end (top)

Presented By:
Sheikh Muhammad Aamir

3 STACK
4 Various Types of Stacks
5 LIFO
6 Postponed Decisions
7 Postponed Decisions...

Click to add notes

Postponed Decisions

- Stacks are frequently used to indicate the order of the processing of data when certain steps of the processing must be postponed until other conditions are fulfilled.

- Suppose we are working on Project–A, while working we need working on Project–B in order to complete the Project–A. Then we place the folder containing the data of A onto a Stack and begin to process Project–B. However, suppose that while processing Project–B we led to Project–C, for the same reason and similar process for Project–D.

Presented By:
Sheikh Muhammad Aamir

Postponed Decisions...

Study Point

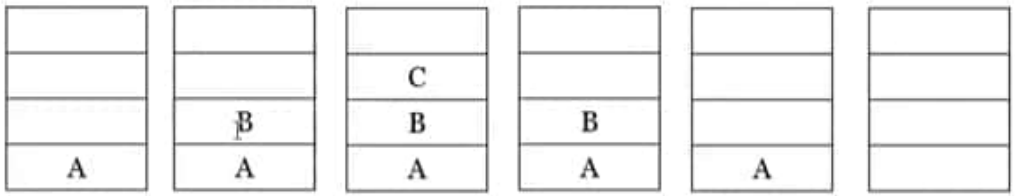
- On the other hand, suppose we are able to complete the processing of D. Then the only project we may continue to process is Project–C, which is on the top of the Stack. Hence, we remove C from the stack and continue to process C.

- Similarly, after completing the processing of C, we remove B from the stack and continue the process B. Finally, after completing the processing of B, we remove the last folder of A from the stack, leaving the Stack empty and continue the processing of our original Project–A.

Presented By:
Sheikh Muhammad Aamir

Postponed Decisions...

Study Point



Presented By:
Sheikh Muhammad Aamir

Click to add notes

Study Point

Postponed Decision...

$n! = n\,(n-1)!$

$(n-1)! = (n-1)\,(n-2)!$

$4! = 4\,(4-1)! = 4 * 3!$

$3! = 3\,(3-1)! = 3 * 2!$

$2! = 2\,(2-1)! = 2 * 1!$

$1! = 1$

$2! = 2 * 1 = 2$

$3! = 3 * 2 = 6$

$4! = 4 * 6 = 24$

Presented By:
Sheikh Muhammad Aamir

Click to add notes

Operations on Stack

Study Point

- Push operation for Insertion
- Pop operation for Deletion

Presented By:
Sheikh Muhammad Aamir

Push Method

Study Point

- push() method adds an element at the top the stack.
- It takes as argument an Object to be pushed.

- It first checks if there is room left in the stack. If no room is left, it throws an exception. Otherwise, it puts the object into the stack array, by incrementing TOP variable by one.

Presented By:
Sheikh Muhammad Aamir

Pop Method

Study Point

- The pop method removes an item from the stack and returns that item.
- The pop method first checks if the stack is empty. If the stack is empty, it throws an exception like Stack is empty. Otherwise, it simply decreases TOP by one and returns the item found at the top of the stack.

Presented By:
Sheikh Muhammad Aamir

Click to add notes

Array Representation of Stacks

Study Point

- Stack is maintained by a linear array, an indicator TOP, which contains the location of top element of the stack; and a variable N which represent the maximum size of array STACK.
- The TOP = 0 or TOP = NULL will indicate that the stack is empty.

Presented By:
Sheikh Muhammad Aamir

Another Example of Stack

Study Point

Stack with size N = 7
TOP = 0 (indicates the Stack is empty)

1	2	3	4	5	6	7

PUSH(25)
TOP

1	2	3	4	5	6	7
25						

PUSH(10)
PUSH(50)
PUSH(75)

TOP

1	2	3	4	5	6	7
25	10	50	75			

Stack operations using array...

Study Point

POP( ) [returns 75 from TOP]

TOP

1	2	3	4	5	6	7
25	10	50				

POP( ) [returns 50 from TOP]

TOP

1	2	3	4	5	6	7
25	10					

PUSH(55) [insert 55 into the stack]

TOP

1	2	3	4	5	6	7
25	10	55				

Presented By:
Sheikh Muhammad Aamir

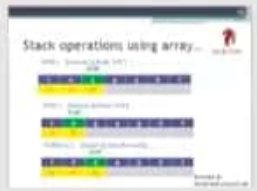
Push Procedure

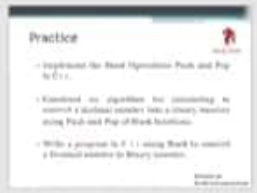
Study Point

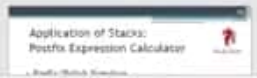
Push (STACK, TOP, X, N)

1. If TOP > = N Then [Is stack already filled ?]
   Write : Stack is Full
   Else:
   Set TOP := TOP +1 [Increasing TOP by 1]
   Set STACK[TOP] := X [Insert X item at new TOP position]
   [End of If structure]
2. Return

Presented By:
Sheikh Muhammad Aamir







17

Pop Procedure

Study Point

POP (STACK, TOP,N)

1. If TOP = 0 Then [Is stack already empty ?]
   Write: Stack Empty
   Return
   Else:
   Set X := STACK[TOP] [Assigns TOP element to X]
   Set TOP := TOP – 1 [TOP decreases by 1]
   Return X
   [end of if structure]

Presented By:
Sheikh Muhammad Aamir

Click to add notes

Practice

Study Point

- Implement the Stack Operations Push and Pop in C++.
- Construct an algorithm for calculating to convert a decimal number into a binary number using Push and Pop of Stack functions.
- Write a program in C ++ using Stack to convert a Decimal number to Binary number.

Presented By:
Sheikh Muhammad Aamir

Click to add notes

19


20


21


22


23


Applications of Stack:

Arithmetic Expressions Solving

Study Point

- Let Q be an expression that contains
  - operands and operators
- Expression can be written in three format
  - Infix Notation
  - Prefix Notation
  - Postfix Notation
- Evaluation of Q by using reverse Polish (Postfix) notation
- Prefix Expression is known as Polish Notation
- Suffix/Postfix Expression is known as Reverse Polish Notation

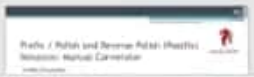
Presented By:
Sheikh Muhammad Aamir

19

20

21

22

23


Study Point

Expressions

- Infix Notation

  e.g. A + B
- Prefix Notation (Polish Notation)

  e.g. + AB
- Postfix Notation (Reverse Polish Notation)

  e.g. AB +

Here A, and B are operands whereas + is an operator

Presented By:
Sheikh Muhammad Aamir

Click to add notes

21

Operator Precedence

Study Point

- Highest: Exponentiation (^ , $, ↑)
- Next Highest: Multiplication (*) and Division (/)
- Lowest: Addition (+) and Subtraction (-)

- All these precedence for binary operations

Presented By:
Sheikh Muhammad Aamir

Click to add notes

Expression Evaluation: Example

Study Point

$$2 \wedge 3 + 5 * 2 \wedge 2 - 12 / 6$$
$$= 8 + 5 * 4 - 12 / 6$$
$$= 8 + 20 - 2$$
$$= 28 - 2$$
$$= 26$$

Presented By:
Sheikh Muhammad Aamir

Prefix / Polish and Reverse Polish (Postfix) Notation: Manual Conversion

Study Point

Prefix Examples

A + B	(A + B) * C	A + (B*C)	(A+B) / (C – D)
=+AB	= +AB * C = *+ABC	= A + *BC = + A*BC	= +AB / -CD = /+AB-CD

Postfix Examples

A + B	(A + B) * C	A + (B*C)	(A+B) / (C – D)
=AB+	= AB+ * C = AB+C*	= A + BC* = ABC*+	= AB+ / CD- = AB+CD-/

Presented By:
Sheikh Muhammad Aamir

Manual Infix to Prefix and Postfix Conversion ...

Study Point

Infix Expression	Prefix Expression	Postfix Expression
(A+B) * (C-D)	= +AB * - CD	= AB+ * CD-
	= *+AB – CD	= AB+ CD- *
A + B * C – (D $ 2 * 3)	A +B * C – ($ D2 * 3)	A +B * C – (D2 $ * 3)
	=A +B * C – (*$ D2 3)	=A +B * C – (D2 $ * 3)
	=A + *B C – (*$ D2 3)	=A +B C * – D2 $ 3 *
	=+A *B C – *$ D2 3	=A B C * + – D2 $ 3 *
	= - +A *B C *$ D2 3	=A B C * + D2 $ 3 * -

Presented By:
Sheikh Muhammad Aamir

Study Point

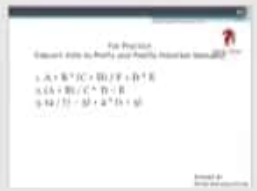
For Practice

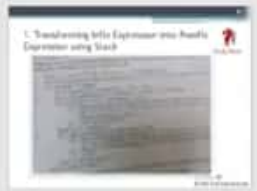
Convert Infix to Prefix and Postfix Notation Manually

1. A + B * (C + D) / F + D * E
2. (A + B) / C * D – E
3. 12 / (7 – 3) + 2 * (1 + 5)

Presented By:
Sheikh Muhammad Aamir

Click to add notes





Expression Solving using STACK

Study Point

1. Infix to Postfix Conversion using STACK
2. Evaluation of Postfix Expression using STACK

Presented By:
Sheikh Muhammad Aamir

Click to add notes

1. Transforming Infix Expression into Postfix Expression using Stack

Study Point

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

1. Push "(" onto 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 STACK.
5. If an operator ⊗ is encountered, then:
   (a) Repeatedly pop from STACK and add to P each operator (on the top of STACK) which has the same precedence as or higher precedence than ⊗.
   (b) Add ⊗ to STACK.
   [End of If structure.]
6. If a right parenthesis is encountered, then:
   (a) Repeatedly pop from STACK and add to P each operator (on the top of STACK) until a left parenthesis is encountered.
   (b) Remove the left parenthesis. [Do not add the left parenthesis to P.]
   [End of If structure.]
[End of Step 2 loop.]

Presented By:
Sheikh Muhammad Aamir

Transforming Infix Expression into Postfix Expression using Stack : Example

- Infix Expression will be as : A * (B + C) –D / E)

Study Point

Now reading last symbol

Sr.	Scanned Symbol	STACK	Expression P
1		(	
2	A	(	A
3	*	(*	A
4	(	(* (	A
5	B	(* (	A B
6	+	(* (+	A B
7	C	(* (+	A B C
8	)	(*	A B C +
9	-	(-	A B C + *
10	D	(-	A B C + * D
11	/	(- /	A B C + * D
12	E	(- /	A B C + * D E
13	)		A B C + * D E / -

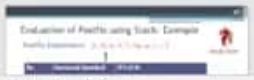
Click to add notes

Transforming Infix Expression into Postfix Expression using Stack: Practice Questions

Study Point

1. A / B – C + D * E – A * C
2. (A + B) / C * D – E
3. A + B * (C ^ D – E)
4. (A – B) * (A + B)
5. 2 * 3 + 5 * 4 - 9

Presented By:
Sheikh Muhammad Aamir



Expression Solving using STACK

Study Point

1. Infix to Postfix Conversion using STACK
2. Evaluation of Postfix Expression using STACK

Presented By:
Sheikh Muhammad Aamir

Click to add notes

2. Evaluation of a Postfix Expression using Stack

Study Point

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

1. Add a right parenthesis ")" at the end of P [This acts as a sentinel.]
2. Scan P from left to right and repeat Steps 3 and 4 for each element of P until the sentinel ")" is encountered.
3. If an operand is encountered, put it on STACK.
4. If an operator ⊗ is encountered, then:
   (a) Remove the two top elements of STACK, where A is the top element and B is the next-to-top element.
   (b) Evaluate B ⊗ A.
   (c) Place the result of (b) back on STACK.
   [End of If structure.]
   [End of Step 2 loop.]
5. Set VALUE equal to the top element on STACK.
6. Exit

Presented By:
Sheikh Muhammad Aamir

Evaluation of Postfix using Stack: Example

Study Point

Infix Expression (Q): 5*(6+2)-(12/4)

Postfix Expression (P): 5 , 6, 2, +, *, 12, 4, /, -

Step 1: Place) at the end of P

Postfix Expression (P): 5 , 6, 2, +, *, 12, 4, /, -,)

Presented By:
Sheikh Muhammad Aamir

Evaluation of Postfix using Stack: Example

Postfix Expression: 5 , 6, 2, +, *, 12, 4, /, -,)

Sr.	Scanned Symbol	STACK
1	5	5
2	6	5, 6
3	2	5, 6, 2
4	+	5, 8
5	*	40
6	12	40, 12
7	4	40, 12, 4
8	/	40, 3
9	-	37
10	)	

Presented By:
Sheikh Muhammad Aamir

Evaluation of Postfix Expression using Stack: Practice Questions

Study Point

1. 2, 3, 1, *, +, 9, -
2. 7, 2, 3, *, 5, +, 8, 4, 2, /, -, *, -
3. A, B, C, D, 2, ^, *, +, 3, +, +

Presented By:
Sheikh Muhammad Aamir