DATA STRUCTURE (DS)

Course Details

COURSE NAME: DATA STRUCTURE using C

What background knowledge is necessary?
Basic knowledge on C is required.

Course Overview

- ✓ *Starts from recursive function, pointer as implementation is based these concepts*
- ✓ *Pseudo-codes, objectives questions and answer at end of every chapter*
- ✓ *Hands on goes on parallel (with individual laptop)*
- ✓ *Explains every algorithm simply and implements it in class. Implementation is more important than only understanding it or getting the codes*
- ✓ *Provides a great practise session*
- ✓ *Discusses time complexity and space complexity for every implementation*
- ✓ *Discussion of frequently asked coding question*

DATA STRUCTURE: Course Details

Course Duration	***50 hrs***
Course Fee	***3,500***
Mode of Teaching	***Live online class***
Class Timing	***10.30am – 12.30pm***
Batch Size	***40-50***
Educator	***1.Mr. Dambarudhar Mohanty (M- 8249431423)*** ***2.Mr. Sudhansu Dalai (M- 9777036556)***

Data Structure :Detailed Syllabus

1. Introduction to Data Structure
2. Algorithm
3. Performance Analysis
   - ✓ Time complexity
   - ✓ Space complexity
4. Asymptotic Notations-
   - ✓ Big O
   - ✓ Omega
   - ✓ Theta notations
5. Stacks
   - ✓ Stack Operations
   - ✓ push()
   - ✓ pop()
   - ✓ isEmpty()
   - ✓ isFull()
   - ✓ Stack implementation using arrays
   - ✓ Applications
   - ✓ String reverse
   - ✓ Stack and Recursion
   - ✓ Balanced Parentheses

6. Stack Implementation using pointer (Dynamic)

7. Arithmetic Expression
    - ✓ Introduction to Notations
    - ✓ Importance of Notations in expression evaluation
    - ✓ Conversion Algorithms
    - ✓ Infix to prefix
    - ✓ Infix to postfix
    - ✓ Evaluation of Postfix expression
    - ✓ ***Implementation of all the conversions (7+)***
    - ✓ *MCQs/ Pseudo Code*
8. Queues
    - ✓ Operations on Queue – enqueue(), dequeue()
    - ✓ Queue implementation using static arrays
    - ✓ Applications
    - ✓ Queues Implementations using pointer (dynamic)
    - ✓ *MCQs/ Pseudo Code*
9. Circular queues
10. Double Ended queue (Deques)
11. Single linked list
    - ✓ Introduction
    - ✓ Construction
    - ✓ Length
    - ✓ Insertion
    - ✓ Deletion
    - ✓ Sort
    - ✓ Reverse list
    - ✓ Swap nodes
    - ✓ Applications
12. Stack implementation using linked list
13. Queue implementation using linked list
14. Doubly linked list
15. Circular linked list
16. Binary Tree
    - ✓ Terminology
    - ✓ Differences between Tree and Binary Tree
    - ✓ Binary Tree Representations
    - ✓ Expression Trees
    - ✓ Traversals
    - ✓ In-order
    - ✓ pre-order
    - ✓ post-order
17. Binary Search Tree
    - ✓ Introduction to BST
    - ✓ Insertion
    - ✓ Deletion
    - ✓ Search
    - ✓ Implementation
    - ✓ *Program (25 +)*
    - ✓ *MCQs/ Pseudo Code*
18. AVL Trees
    - ✓ Introduction
    - ✓ BST v/s AVL
    - ✓ Rotations
    - ✓ L-L-Rotation
    - ✓ R-R-Rotation

   - L-R-Rotation
   - R-L-Rotation
   - Insertion
   - Deletion
   - Implementation
   - Traversal
   - *MCQs/ Pseudo Code*

19. B trees
    - M-way Search Tree
    - Search
    - Insertion
    - Deletion
20. Searching Algorithms
    - Linear search
    - Binary search
21. Sorting Algorithms
    - Bubble sort
    - Selection sort
    - Insertion sort
    - Heap sort
    - Merge sort
    - Quick sort
    - Count sort
    - Radix sort
    - *MCQs/ Pseudo Code*
22. Graph
    - Introduction & Terminology
    - Graph Representations
    - Dijkstra's Algorithm
    - Kruskal's Algorithm
    - Traversal
    - BFS (Breadth First Search)
    - DFS (Depth First Search)
    - *MST*
    - *MCQs/ Pseudo Code*
23. Hashing
    - Hash Table representation
    - Hash function-Division Method
    - Collision
    - Collision Resolution Techniques
    - Separate Chaining
    - Open addressing
    - Linear probing
    - Quadratic probing
    - Double hashing
    - Rehashing
    - *Programs (5+)*

Educator:

1. **Mr. Dambarudhar Mohanty (M- 8249431423)**
2. **Mr. Sudhansu Dalai (M- 9777036556)**