

Data Structures (using C)

By Surya Teja Achanta

Course 1: Introduction to Structures, Dynamic Memory Allocation (By June 28).

1. Course overview.
2. Prerequisites for the course.
3. Random number and its implementation.
4. Introduction to Data Structures.
5. Understanding Structures and creating structures.
6. Dynamic Memory Allocation and Implementation.
7. Implementation of Triangle using structures, Dynamic memory Allocation.
8. Finding area of Triangle and implementing some other functions.
9. Implementation of Circle using structures, Dynamic memory Allocation.
10. Finding area, overlaps, tangents (any two) of a circle.

Course 2: linked list, stack and Queue (By July 6).

1. Introduction to Linked list, stack and Queue.
2. Creating and traversing the linked list.
3. Implementing Stacks and Queues using linked List.
4. Printing the data in reverse order.
5. Function for reversing of Data in a Linked List
6. Sorting of Linked List and printing the sorted linked list.
7. Comparing two Linked lists.
8. Merging of two sorted linked list.
9. Deleting of a node in the Linked list.
10. Deleting Duplicate node from a sorted linked list.

11.Cycle detection of a linked list.

Course 3:Trees and Heaps(By July 15).

- 1.Course overview.
- 2.Introduction to Trees.
- 3.Implementation of Binary Tree.
- 4.Tree Traversals.
- 5.Implementing Breadth First Traversal.
- 6.Implementing Depth First Traversal.
- 7.Finding maximum and minimum value of a tree.
- 8.Implementation of Binary Search Tree.
- 9.Finding Height of a Given Binary Tree.
- 10.How to print the top view of a Binary Tree.
- 11.Finding Mirror of a given tree.
- 12.Inserting,Deleting and Searching of Data in trees.
- 13.Introduction to heaps.
- 14.Implementing Heaps(part1) .
- 15.Implementing Heaps(part 2).
- 16.Dfs traversal of a heap.
- 17.Implementing Some Important programmes of Heaps.

Course 3:Graphs(Adjacency Matrix and Adjacency List)(by July 22).

- 1.Course Overview.
- 2.Introduction to graphs.
- 3.Implementation of graph using adjacency matrix(part 1).

4. Implementation of graph using adjacency matrix(part 2).
- 5.Graph traversal in adjacency matrix.
- 6.Finding degree of each vertices in adjacency matrix.
- 7.Implementaton of DFS.
- 8.Implementation of graph using adjacency list(part 1).
- 9.Implementation of graph using adjacency list(part 2).
- 10.Graph traversal in Adjacency list.
- 11.Construction of Spanning tree of Implemented adjacency list.
- 12.Finding degree of each vertices in an adjacency list.
- 13.Shortest path algorithm(b/w any two vertices).