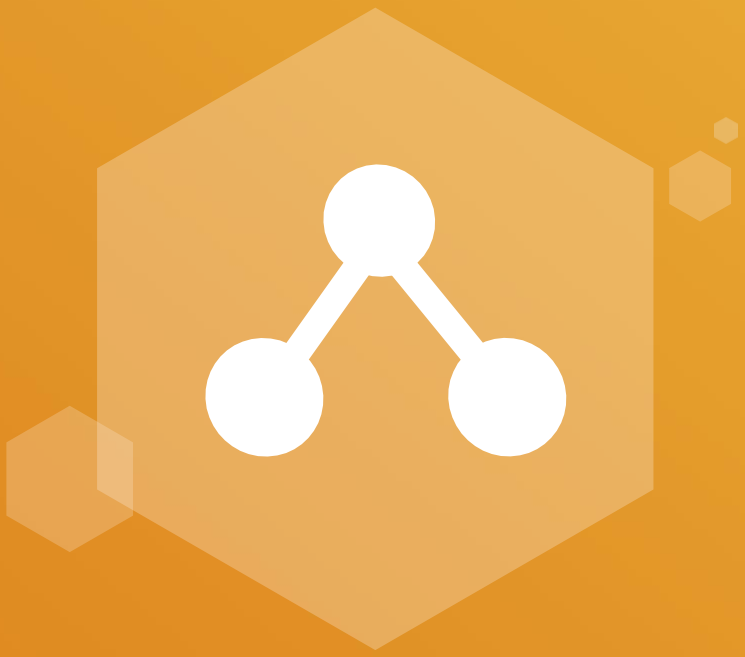
GeeksforGeeks

Data Structures

With Python



Detailed
Course Syllabus

1. Analysis of Algorithms

a. Analysis of Algorithms(Background)
b. Order of Growth
c. Best, Average and Worst Cases
d. Big O Notation
e. Omega Notation
f. Theta Notation
g. Analysis of CommonLoops
h. Analysis of Recursion
i. Space Complexity

2. List in Python

a. List (Dynamic Sized Array) Introduction
b. Working of List in Python
c. Average or Mean of A List
d. Separate Even andOdd
e. Get Smaller Elements
f. Slicing (List,Tuple And String)
g. Comprehensions in Python
h. Largest Element in a List
i. Second Largest Element in a list
j. Check if a list is Sorted
k. Find the only Odd
l. Reverse a List in Python
m. Left Rotate a List by one
n. Left Rotate by d Places

3. String

a. Escape Sequences and Raw Strings
b. Reverse A String in Python
c. String Comparison in Python
d. Strings in Python
e. Formatted String in Python

f. String Operations Part (1)
g. String Operations Part (2)
h. Pattern Searching in Python
i. Check For Palindrome In Python
j. Check For Anagram in Python

4. Hashing

a. Introduction to Hashing
b. Hashing Application
c. Direct Address Table
d. Hashing Functions
e. Collision Handling
f. Chaining
g. Implementation of Chaining in Python
h. Open Addressing
i. Double Hashing
j. Implementation of Open Addressing in Python
k. Chaining vs Open Addressing
l. Set in Python
m. Dictionary in Python
n. Count Distinct Elements in a List
o. Subarray with 0 sum in python
p. Check for Palindrome Permutation

5. Recursion

a. Recursion in Python
b. Applications of Recursion
c. Practice For Recursion (Part 1)
d. Practice For Recursion (Part 2)
e. Sum of Natural Numbers Using Recursion
f. Print N to 1 using Recursion in Python
g. Print 1 to N using Recursion in Python
h. Sum Of Digits Using Recursion
i. Tower of Hanoi in Python
j. Josephus Problem in Python

6. Searching

a. Binary Search in Python
b. Recursive Binary Search in Python
c. Analysis of Binary Search
d. Index of First Occurrence
e. Index of Last Occurrence
f. Count Occurrences in a Sorted Array

7. Sorting

a. Bubble Sort
b. Selection Sort
c. Insertion Sort inPython
d. Merge Two SortedArrays
e. Merge Subarrays
f. Merge Sort Algorithm
g. Merge Sort Analysis
h. Quick Sort Introduction
i. Partition a Given Array
j. Lomuto Partition
k. Hoare's Partition
l. Quick Sort using Lomuto Partition
m. Quick Sort using Hoare's Partition
n. Analysis of Quick Sort
o. Space Analysis of Quick Sort
p. Tail Call Elimination in Quick Sort
q. Sorting in Python
r. List Sort inPython
s. Sorted in Python

8. Linked List

a. Linked List Introduction in Python
b. Simple Linked List Implementation in Python
c. Applications of LinkedList
d. Traversing a Linked List in Python
e. Search in LinkedList
f. Insert At The Beginning of Linked list in Python
g. Insert at The End Of Linked List
h. Insert at Given Position in Singly Linked list
i. Delete First Node Of Linked List in Python
j. Delete Last Node of Linked List
k. Sorted Insert Linked List in Python
l. Reverse a Linked List In Python
m. Recursive Reverse A Linked List (Part 1)
n. Recursive Reverse A Linked List (Part 2)

9. Circular Linked List

a. Circular Linked List in Python
b. Circular Linked List (Advantages & Disadvantages)
c. Circular Linked List traversal
d. Insert at the Beginning of Circular Linked List
e. Insert at The End of A Circular Linked List
f. Delete Head of circular Linked List
g. Delete Kth Node of Circular Linked List

10. Doubly Linked List

a. Doubly Linked List in Python
b. Singly Vs Doubly Linked List (Advantages & Disadvantages)
c. Insert at the Beginning of DLL in Python
d. Insert at the End of DLL in Python
e. Delete Head of A Doubly Linked List
f. Delete Last Node of DLL in Python
g. Reverse A Doubly Linked List in Python

11. Stack

a. Stack Data Structure
b. Stack Applications
c. Stack in Python
d. Linked List Implementation of Stack in Python
e. Check for Balanced Parentheses in Python
f. Infix, Prefix and Postfix Introduction
g. Infix to Postfix (Simple Solution)
h. Infix to Postfix (Efficient Solution)
i. Evaluation of Postfix
j. Infix to Prefix (Simple Solution)
k. Infix to Prefix (Efficient Solution)
l. Evaluation of Prefix

12. Queue

a. Queue Data Structure
b. Application of Queue Data structure
c. Queue in Python
d. Linked List Implementation of Queue in Python
e. Queue Implementation using Circular List

13. Tree

a. Tree Data Structure
b. Tree Traversal
c. Application of Tree
d. Binary Tree in Python
e. Inorder Traversal in Python
f. Preorder Traversal in Python
g. Postorder Traversal in Python
h. Size of Binary Tree in Python
i. Maximum in Binary Tree
j. Search Binary Tree
k. Height of Binary Tree
l. Iterative Inorder Traversal
m. Iterative Preorder Traversal
n. Level Order Traversal

14. Binary Search Tree

a. Binary Search Tree(Background)
b. Binary Search Tree in Python
c. Search in BST in Python
d. BST insert in Python
e. BST Delete in Python
f. Floor in BST (Problem and Solution Idea)
g. BST Floor in Python
h. Ceiling in BST in Python

15. Heap

a. Binary Heap Introduction
b. Heap Python Implementation(Introduction)
c. Binary Heap Insert
d. Binary Heap (Extract min and Heapify)
e. Decrease Key and Delete Operations
f. Build Heap
g. Heapq in Python

16. Graph

a. Introduction to Graph
b. Graph Representation (Adjacency Matrix)
c. Graph Representation (Adjacency List)
d. Graph Adjacency List Representation in Python
e. Adjacency Matrix and List Comparison
f. Breadth First Search in Python
g. BFS for Disconnected Graph
h. Connected Components in an Undirected Graph using BFS
i. Applications of BFS
j. Depth First Search
k. DFS For Disconnected Graph
l. Connected Components in an Undirected Graph using DFS
m. Applications of DFS