

COURSE TITLE: Data Structures and Algorithms-I

-----

A Core Course (UG1) for CSE/ECE

Course Level: Core

L-T-P-C: 2 - 2 - 3 - 4

Prerequisites: 1. Intro to C programming 2. Overview of Computers

Note: The course number and name might vary based on the semester/year.

**Outline:** Topics include elementary data structures, (including arrays, stacks, queues, and lists), advanced data structures (including trees and graphs), the algorithms used to manipulate these structures

**Course Objectives:** To assess how the choice of data structures and algorithm design methods impacts the performance of programs. To choose the appropriate data structure and algorithm design method for a specified application. To write programs using procedure-oriented design principles.

**COURSE OUTLINE (TOPICS):** The following list of topics is very tentative.

Based on available time slots, some topics may be dropped or added or reordered. No. of hours may vary as per the requirement of any particular topic(s).

**Unit - I** (No of hours: 7)

**C REFRESHER and ALGORITHMIC ANALYSIS**

**ARRAYS AND LINKED LISTS:** Storage structures for arrays, matrix, row-major, column-major, Sparse matrices. Linked list, Doubly linked lists, Circularly linked lists, Dynamic storage management - Garbage collection and compaction.

**Unit - II** (No of hours: 7)

**STACK AND QUEUE:** Stacks and queues: insertion, deletion, Stack and queue using linked list, Circular queue, Prefix, postfix, infix notation and conversions.

**Unit - III** (No of hours: 7)

**TREES:** Binary tree insertion, deletion, traversal (inorder, preorder and postorder), Binary Search Tree, Threaded binary tree, AVL tree, Extended Topics: B-tree, B+-tree (if time permits).

**Unit - IV** (No of hours: 7)

**SORTING AND SEARCHING:** Selection sort, Insertion sort, Bubble sort, Merge Sort, Heap sort, and Quick sort, sorting in linear time, Hash Tables.

**Unit - V** (No of hours: 7)

**GRAPH:** Representation of Graphs, Breadth First Search, Depth First Search, Topological Sort, Strongly Connected Components, Algorithm for Kruskal's and Prim's for Finding Minimum cost Spanning Trees, Dijkstra's Algorithm for finding Single source shortest paths.

#### **TEXTBOOKS**

- Seymour Lipschutz, Data Structures with C, Schaum's Outline Series
  - Langsam Yedidiah, Augenstein J Moshe, Tenenbaum M, Data Structures using C and C++, PHI.
- Weiss, Mark Allen. *Data Structures and Algorithm Analysis in C. Second Edition*. Pearson Education India. **(DSAC)**

#### **REFERENCE BOOKS**

- Horowitz, Sahni, Freed, Fundamentals of Data Structures in C, Silicon Press.
- Kruse R., Data Structures and Program Design in C, Pearson Education India.
- Cormen et al. Introduction to Algorithm 3rd Edition. MIT Press, 2017.

#### **COURSE WORK:**

Course grades will be based on the following weightage pattern. This is purely indicative and necessary modification will be made by the faculty depending on the intensity of the different work assigned for various tasks.

a) EXAMINATIONS: 50%

Mid Semester - 1 Exam: 15%

Mid Semester - 2 Exam: 15%

End Semester Exam: 20%

b) Assignments: 30%

c) Lab Exam: 20%

**Industrial Relevance:** Used in all software industries and significant role in tech giants like Google, Facebook, Amazon, Flipkart where interviews are more focused on measuring the knowledge of Data Structures and Algorithms of the candidates.

**Companies:** <https://www.naukri.com/data-structures-jobs>  
Tech giants like Amazon, Microsoft, Google, Flipkart, Walmart, Amadeus, Facebook to name a few

#### **Similar Courses:**

<http://www.cse.iitd.ernet.in/~naveen/courses/CSL201/index.html>

**Course Ethics:** 1. We expect students to work alone on all programming assignments. There are no group projects or assignments in this course.

2. Using the internet for help with syntax and compilation errors will be tolerated. But, a student should never copy paste a solution (or partial solution) from any online source.

3. Plagiarism in any shape or form will not be tolerated and will be dealt with very severely.