

Gujarat Technological University

Master of Computer Applications

Semester-II

Subject Name: **Data Structures (DS)**

Subject Code: **2620001**

Objectives:

- To develop proficiency in the specification, representation, and implementation of Data Types and Data Structures.
- To be able to carry out the Analysis of various Algorithms for mainly Time and Space Complexity.
- To get a good understanding of applications of Data Structures.
- To develop a base for advanced computer science study.

Prerequisites:

Any programming language like C, C++

Contents:

- 1. Analysis of algorithms :** [6 Lectures]
Examples of Insertion Sort, Pseudo Code Convention, Designing Algorithms - Example of Merge Sort , Growth of Functions - Asymptotic Notation
- 2. Introduction to Data Structures :** [4 Lectures]
Primitive Data Structures, String Manipulation & Pattern Matching, Storage Representation of Strings , Text Handling , KWIC Indexing
- 3. Linear Data Structures:** [14 Lectures]
Arrays, Storage Structure for Arrays, Structures & Arrays of Structures , Stack, Applications of Stacks, Queues, Simulation, Priority Queues, Pointers & Linked Allocation , Linked Linear Lists , Circularly Linked Linear Lists , Doubly Linked Linear Lists, Applications of Linked Linear Lists
- 4. Nonlinear Data Structures:** [14 Lectures]
Trees , Operations on Binary Trees , Storage Representation & Manipulation of Binary Trees, Conversion of General Tree to Binary Trees , Sequential & Other Representation of Trees , Application of Trees - Manipulation of Arithmetic Expression , Multi-linked Structures - Sparse Matrices , Matrix Representation of Graphs , Graphic Representation of List Structures , Other Representation of Graphs , Breadth First Search (BFS) , Depth First Search (DFS) , Spanning Trees , Garbage Collection
- 5. Sorting & Searching:** [12 Lectures]
Introduction , Selection Sort , Bubble Sort , Merge Sort , Heap Sort , Quick Sort , Radix Sort , Sequential Searching , Binary Searching , Search Trees - Height Balanced , 2-3 Trees , Weight Balanced , m-ary Trees , Trie Structures , Hash Table Search Methods , Introduction , Hashing Functions , Collision Resolution Techniques

Text Books:

1. "An Introduction to Data Structures with Applications", Jean-Paul Tremblay, Paul G. Sorenson, Tata McGraw-Hill, 2nd Edition, (2007)
2. "Introduction to Algorithm", Cormen, Leiserson, Rivest, Stein, , PHI (2003), 2nd Edition,
3. "Design and Analysis of Algorithms" Parag Dave & Himanshu Dave, Pearson Education (2008).

Reference Books:

1. "Data Structures and Algorithm Analysis in C", Mark Allen Weiss, , Pearson Education.
2. "Data Structures: A Pseudo-code Approach with C", Gilberg & Forouzan, , Cengage Learning.
3. "Data Structures Via C++: Objects by Evolution", A. Michael Berman, , Oxford Univ. Press (2004)
4. "Fundamentals of Data Structures in C", Horowitz, Sahni, Anderson-Freed, , University Press (2nd edition-2007)
5. "Data Structures Using C & C++", Tenenbaum, PHI.
6. "Data Structures & Algorithms" , A V Aho, J E Hopcroft, J D Ullman, , Pearson Education (1983).
7. "Sorting & Searching - The Art of Computer Programming" D E Knuth, , Vol. 3, Pearson Education (1998).
8. "Data structures and algorithms, concepts, Techniques and Applications" ,G. A.V. PAI, , TMH , 1st Edition (2008)
9. "Algorithm design-foundation, analysis & internet examples", Michel Goodrich, Roberto Tamassia, , Wiley

Chapter wise Coverage from Text Books:

Book-1 :

1.4, 2.1, 2.2.2, 2.2.3, 2.4, 2.5.1, 2.5.3
3.1 to 3.8, 4.1, 4.2.1, 4.2.2, 4.2.3, 4.3
5.1.1, 5.1.2, 5.1.3, 5.1.4, 5.1.5, 5.2.1, 5.3.1, 5.4.1, 5.4.2, 5.4.3, 5.4.4, 5.4.5, 5.4.6, 5.6.5
6.1.1, 6.1.2, 6.1.3, 6.1.4, 6.1.5, 6.1.6, 6.1.7, 6.2.1, 6.2.2, 6.2.3, 6.2.3.1, 6.2.3.2, 6.2.3.3, 6.2.3.4, 6.2.4, 6.2.4.1, 6.2.4.2, 6.2.4.3

Book-2

2.1, 2.2, 2.3, 3.1

Book -3

4.4, 4.5

Accomplishments of the student after completing the Course:

- Ability to decide the appropriate data type and data structure for a given problem.
- Ability to select the best algorithm to solve a problem by considering various problem characteristics, such as the data size, the type of operations, etc.
- The algorithms as referred above would include various operations on Queues, Stacks, Linked Lists, Trees, Graphs, Sorting, Searching, Hash tables
- Ability to compare algorithms with respect to time and space complexity