

# 10B11CI211: Data Structure

**Course Credit: 4**

**Semester: II**

## Introduction

Data Structures is a core module in the field of computer science. Different Data Structures are heavily used in most of the software whether it is system software or application software. The field of compiler designing, database management systems, operating systems, searching and sorting heavily depends on Data Structures.

In this module students will be taught the basic concepts of Information storage and retrieval using various available Data Structures, their implementation using C language, time and space complexities of algorithms.

## Course Objectives (Post-conditions)

### Knowledge objectives:

- Develop problem solving ability using Programming.
- Develop ability to design and analyze algorithms.
- Introduce students to data abstraction and fundamental data structures.
- Develop ability to design and evaluate Abstract Data Types and data structures.
- Apply data structure concepts to various examples and real life applications

### Application objectives:

- Write programs that use arrays, records, linked structures, stacks, queues, trees, and graphs.
- Compare alternative implementations of data structures with respect to performance .
- Compare and contrast the benefits of dynamic and static data structures implementations Design and implement an appropriate hashing function for an application .
- Finding the computational efficiency of the principal algorithms for sorting, searching, and hashing.

## Expected Student Background (Preconditions)

: Introduction to Programming

## Topics Outline:

S NO	Topics	Hrs
1	Introduction to Data Structures, Abstract Data Type (ADT), Arrays and Strings, Structures, Recursion, Pointers, Dynamic memory allocation	5
2	Algorithm Design, Scalability, Introduction to Complexity Analysis , Big O Notation, Relationship between time complexity and hardware performance,	3
3	Linked Lists:- ADT type, Linear List , Linear Linked list, doubly linked list, circular linked list, header Linked list, various implementations and applications of Linked Lists	5

4	Stack: - ADT type, specifications, array based and linked list based, recursion and its removal with stack .	3
5	Queues:- ADT type, array based and linked list based,, queue as buffer, searching, Circular queues, Deque, Managing multiple queues, , various implementations and applications of Queues	4
6	Binary Trees:- Introduction to non-linear data structures, ADT type, array based and linked list based, binary tree, binary search tree, AVL tree, tree traversal,	6
7	Sorting Algorithms: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort,	8
8	Searching:- Linear and Binary Search Hashing:- Hash table	3
9	Graphs: - ADT type, array based and linked list based, graph traversal algorithms i.e. Breadth First & Depth First, various implementations and applications of graphs	5
	Total	42

### References

1. Langsam, Augestein, Tenenbaum : Data Structures using C and C++, 2nd Edn, 2000,
2. Horowitz and Sahani: Fundamental of Data Structures in C, 2<sup>nd</sup> Edn, 2008
3. Kruse, Tonso, Leung: Data Structures and Program Design in C, 2000
4. Richard F. Gilberg & Behrouz Forouzan: Data Structures, A Pseudocode Approach with C, 2001.
5. Weiss: Data Structures and Algorithm Analysis in C/C++, 3<sup>rd</sup> Edn, 2006
6. Carrano and Prichard: Data Abstraction and Problem solving with C++, 5<sup>th</sup> Edn, 2007
7. Sedgewick : Algorithms in C/C++
8. Sahani : Data Structures, Algorithms and applications in C++, 1997.
9. Corman et al: Introduction to Algorithms, 3<sup>rd</sup> Edn., 2009.
10. Heileman : Data Structures, Algorithms and Object Oriented Programming, 2002.
11. Sorenson and Tremblay: An Introduction to Data Structures with Applications, 2<sup>nd</sup> Edn,

**Evaluation Scheme:**

S.No	Examination	Marks
1	T-1	15
2	T-2	25
3	T-3	35
4	*Internal Marks	25

\*Internal Marks Breakdown:

Assignments            9 marks (3x3)

Quizzes                12 marks (3x4)

Regularity            4 Marks