INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

NAME OF DEPT./CENTRE: **Electronics and Computer Engineering**

1. Subject Code: **EC - 251** Course Title: **Data Structures**

2. Contact Hours: **L: 3 T: 1 P: 2**

3. Examination Duration (Hrs.): **Theory** 0 3 **Practical** 0 0

4. Relative Weight: **CWS** 15 **PRS** 15 **MTE** 30 **ETE** 40 **PRE** 00

5. Credits: 0 5 6. Semester ☑ **Autumn** □ **Spring** □ **Both**

7. Pre-requisite: **EC - 101A / EC - 101B**

8. Subject Area: **DCC**

9. Objective: To provide basic data structure concepts in an object-oriented setting for design, implementation, testing and maintenance of software systems.

10. Details of the Course:

Sl. No.	Contents	Contact Hours
1.	**Complexity Analysis:** Time and Space complexity of algorithms, asymptotic analysis, big O and other notations, importance of efficient algorithms, program performance measurement, data structures and algorithms.	3
2.	**Linear Lists:** Abstract data type, sequential and linked representations, comparison of insertion, deletion and search operations for sequential and linked lists, list and chain classes, exception and iterator classes for lists, doubly linked lists, circular lists, linked lists through simulated pointers, lists in STL, skip lists, applications of lists in bin sort, radix sort, sparse tables.	8
3.	**Stacks and Queues: A**bstract data types, sequential and linked implementations, exception handling in classes, representative applications such as parenthesis matching, towers of Hanoi, wire routing in a circuit, finding path in a maze, simulation of queuing systems, equivalence problem.	6
4.	**Hashing:** Search efficiency in lists and skip lists, hashing as a search structure, hash table, collision avoidance, linear open addressing, chains, uses of hash tables in text compression, LZW algorithm.	4
5.	**Trees:** Binary trees and their properties, terminology, sequential and linked implementations, tree traversal methods and algorithms, heaps as priority queues, heap implementation, insertion and deletion operations,	8

	heapsort, heaps in Huffman coding, leftist trees, tournament trees, use of winner trees in mergesort as an external sorting algorithm, bin packing.	
6.	**Search Trees**: Binary search trees, search efficiency, insertion and deletion operations, importance of balancing, AVL trees, searching insertion and deletions in AVL trees, red-black trees, comparison with AVL trees, search insert and delete operations.	4
7.	**Multiway Trees:** Issues in large dictionaries, m-way search trees, B-trees, search insert and delete operations, height of B-tree, 2-3 trees, sets and multisets in STL.	5
8.	**Graphs:** Definition, terminology, directed and undirected graphs, properties, connectivity in graphs, applications, implementation – adjacency matrix and linked adjacency chains, graph traversal – breadth first and depth first, spanning trees.	4
	Total	**42**
	Laboratory component (a) Programming of various data structures and applications in C++ and Java. (b) Data structure programming using STL.	**14x2**

11. Suggested Books:

Sl. No.	**Name of Books / Authors**	**Year of Publication**
1.	Sahni, S., “Data Structures, Algorithms, and Applications in C++”, WCB/McGraw-Hill.	2001
2.	Sahni, S., “Data Structures, Algorithms, and Applications in Java”, WCB/McGraw-Hill.	2001
3.	Drozdek, A., “Data Structures and Algorithms in C++”, Vikas Publishing House.	2002
4.	Wirth, N., “Algorithms and Data Structures”, Prentice-Hall of India.	1985
5.	Lafore, R., “Data Structures and Algorithms in Java”, 2nd Ed., Dorling Kindersley.	2007