Subject Code: 01CT0308

Subject Name: Data Structure using C++

B. Tech. Year – II (Semester III)

Objective: The objective is this course is to teach efficient storage mechanisms of data for an easy access and to design and implementation of various basic and advanced data structures. Further, this course introduces various techniques for representation of the data in the real world and helps in designing and developing application using efficient data structures for protection and management of data and improving the logical ability.

Credits Earned: 04 Credits

Course Outcomes: After completion of this course, student will be able to:

1. Implement object-oriented concepts in C++ (Apply).
2. Differentiate linear and non-linear data structures like stacks, queues, linked list etc (Analyze).
3. Choose appropriate data structure as applied to specified problem definition (Evaluate).
4. Demonstrate operations like searching, insertion, deletion, traversing mechanism etc. on various data structures through programming (Apply).
5. Select appropriate sorting and searching algorithm based on problem definition in order to get optimum solution (Evaluate).
6. Compare and contrast the benefits of dynamic and static data structures implementation (Analyze).

Pre-requisite of course: Basic knowledge of C language, Object Oriented Concepts

Teaching and Examination Scheme:

Teaching Scheme (Hours)			**Credits**	**Theory Marks**			**Tutorial / Practical**		**Total Marks**
				E	I		V	T	
Theory	Tutorial	Practical		ESE	IA	CSE	Viva	Term Work	
3	0	2	4	50	30	20	25	25	150

Contents:

Unit	Topics	Contact Hours
1	**Introduction of C++:** Introduction, Data types, Expression and control statements Iteration statements in C++, Arrays and String, Functions, Structures, Class, Object, Friend Function, Static variables and Functions in class, Constructors and Destructors, Inheritance in C++, Types of Inheritance, Pointers, Virtual Functions, Polymorphism, Abstract classes, Templates in C++, Exception Handling in C++	8
2	**Linked List:** Linked List as an ADT, Linked List Vs. Arrays, and Memory Allocation & De-allocation for a Linked List, Linked List operations, Types of Linked List, Implementation of Linked List, Application of Linked List	8
3	**Stack:** The Stack as an ADT, Stack operation, Array Representation of Stack, Link Representation of Stack, Application of stack – Recursion, Polish Notation	4
4	**Queues:** The Queue as an ADT, Queue operation, Array Representation of Queue, Linked Representation of Queue, Circular Queue, Priority Queue, & Dequeue, Application of Queues – Johnsons Algorithm, Simulation	4
5	**Trees:** Basic trees concept, Binary tree representation, Binary tree operation, Binary tree traversal, Binary search tree (BST) implementation, Thread Binary tree, The Huffman Algorithm, Expression tree, Introduction to Multiway search tree and its creation (AVL, B-tree, B+ tree), AVL tree balancing; B-tree; Application of trees	6
6	**Graphs:** Basic concepts, Graph Representation, Graph traversal (DFS & BFS)	4
7	**Sorting:** Sort Concept, Shell Sort, Radix sort, Insertion Sort, Quick Sort, Merge sort, Heap Sort	4
8	**Searching:** List Search, Linear Index Search, Index Sequential Search, Hashed List Search, Hashing Methods, Collision Resolution (One way and Two way);	4
	TOTAL HOURS	**42**

Suggested Text books / Reference books:

1. Data structures and Algorithms in C++, Michael T.Goodrich, R.Tamassia and .Mount, Wiley student edition, John Wiley and Sons, 2e, 2011.
2. Data Structures, Algorithms and Applications in C++, Sartaj Sahani, 2e, 2004.
3. Object Oriented Programming with C++, E.Balaguruswamy, TMH, 6e, 2013.
4. C++: The Complete Reference- Schildt, McGraw-Hill Education (India), 4e, 2017

Suggested Theory distribution:

The suggested theory distribution as per Bloom's taxonomy is as per follows. This distribution serves as guidelines for teachers and students to achieve effective teaching-learning process.

Distribution of Theory for course delivery and evaluation					
Remember	Understand	Apply	Analyze	Evaluate	Create
15%	20%	25%	15%	15%	10%

Suggested List of Experiments:
Minimum 14 experiments to be performed during the semester

1. Implementations of Linked Lists menu driven program.
2. Implementation of different operations on linked list – copy, concatenate, split, reverse, count no. of nodes etc.
3. Representation of Sparse matrix using multilinked structure. Implementation of sparse matrix multiplication.
4. Implementation of polynomials operations (addition, subtraction) using Linked List.
5. Implementations of Linked Lists menu driven program (stack and queue).
6. Implementations of Double ended queue using Linked Lists.
7. Implementation of Priority queue program using Linked Lists.
8. Implementations of stack menu driven program.
9. Implementation of multitask in one array.
10. Implementations of Infix to Postfix Transformation and its evaluation program.
11. Implementations of Infix to Prefix Transformation and its evaluation program.
12. Simulation of recursion.
13. Implementations of circular queue menu driven program.
14. Implementations of double ended queue menu driven program.
15. Implementations of queue menu driven program.
16. Implementation of Priority queue program using array.
17. Implementation of Johnsons Algorithm.

18. Implementation of Simulation Problem.
19. Implementations of Binary Tree menu driven program.
20. Implementation of Binary Tree Traversal program.
21. Implementation of construction of expression tree using postfix expression.
22. Implementations of Huffman code construction.
23. Implementations of BST program.
24. Implementation of various operations on tree like – copying tree, mirroring a tree, counting the number of nodes in the tree, counting only leaf nodes in the tree.
25. Implementations of B-tree menu driven program.
26. Implementations of B+ tree program.
27. Implementation of Preorder traversal of a threaded binary tree.
28. Implementations of AVL Tree menu driven program.
29. Implementations of Shell sort, Radix sort and Insertion sort menu driven program.
30. Implementations of Quick Sort, Merge sort and Heap Sort menu driven program.
31. Implementations of searching methods (Index Sequential, Interpolation Search) menu driven program.
32. Implementation of hashing functions with different collision resolution techniques.
33. Implementations of Graph menu driven program (DFS & BSF).

Supplementary Resources:

1. http://www.nptelvideos.in/2012/11/programming-and-data-structure.html
2. http://www.nptelvideos.in/2012/11/data-structures-and-algorithms.html
3. http://www.geeksforgeeks.org/data-structures/
4. https://www.hackerrank.com/domains/data-structures/arrays
5. Data Structures And Algorithms Made Easy -To All My Readers By Narasimha Karumanchi