SEMESTER – II

Department of Computer Science & Engineering

SUBJECT: DATA STRUCTURE

Course Code	CS102
Teaching Hours/Week(L-P-T-E)	(3-0-0-0)
Credits	03
Course offering department	Department of Computer Science and Engineering
Total Contact Hours	42

Course objectives:

- The course will provide the basic and fundamental knowledge of various data structure concepts for solving different problems in Computer Science
- It will give an idea to create mathematical modeling of computing problems
- It will make the student understand the relationship between algorithms and programming constructs.

Module–1 (10 hrs.)

Stack And Queue:
Fundamentals of Data Structure, Abstract Data Types: Primitive vs Non-primitive, Linear vs Nonlinear, Static vs Dynamic Data Structures
Stack: Fundamentals of the stack, Array representation of Stack, Applications of the stack: Recursion, conversions of Expression; evaluation of Expression.
Queue: Fundamentals of queue, Array representation of Queue, Circular queues, Double ended queues, Applications of queue.

Module–2 (08 hrs.)

Linked List:
Single linked lists: operations and implementation. Double linked list: operations and implementations. Circular linked list: concepts and implementation. Applications: Stack & Queue implementation using linked list, Polynomial manipulation using linked list.

Module–3 (08 hrs.)

Tree:
Fundamentals of Trees, Array Representation of tree, Linked list representation of Tree, Tree Traversals: Recursive and Non-recursive. Operations on binary tree: Finding Height, Leaf nodes, Counting number of nodes. Construction of binary tree from traversal paths. Binary search tree and operations on it, properties of BST, Binary expression tree. AVL tree and operations on it, m-way search tree and B-tree

Module–4 (08 hrs.)

Graph:
Graphs: Fundamentals of Graph, Representation of graphs (Adjacency Matrix, Adjacency List), Path matrix. Traversals: Breadth First Search (BFS), Depth First Search (DFS). Spanning Trees, Minimum Spanning Trees. Dijkstra's shortest path algorithm, Bellman’s Ford Algorithm.

Module-5 (08 Hrs.)

Sorting & Searching:
Sorting techniques: Bubble Sort, Selection Sort, Insertion Sort, Counting Sort. Heaps: Max Heap, Min Heap, Heap Sort. Searching techniques: Linear and Binary search methods (recursive and non-recursive). Hashing: Different hash functions, Collisions, Collision resolution techniques.

Text Books :

1. "Data Structure: A Pseudo code approach with C",Richard F. Gilberg and Behrouz A. Forouzan, Thomson publication., Second Edition.
2. "Data structure in C", A.M.Tanenbaum, Y. Langsam, M. J. Augenstein, PHI publication / Pearson publication., Seventh Edition.

Reference Books:

1. "Data Structures with C"– (Schaums Series), Seymour Lipschutz Tata-McGraw-Hill., First Edition.
2. "Fundamentals of Data Structure in C"– Horowitz, Sahani& Freed, Computer Science Press., Second Edition.
3. "Data Structure Through C: A Practical Approach" by G. S. Baluja, DhanpatRai Publications., First Edition.
4. "Data Structures & Algorithms; Concepts, Techniques & Algorithms " by Pai, Tata McGraw Hill., First Edition.
5. "Data Structures Using C",ReemaThareja, Oxford University Press, Second Edition.

Open Source:

1. NPTEL video series, Data-structures and Algorithms, Instructor: Naveen Garg. URL: https://nptel.ac.in/courses/106102064
2. MIT open Course ware, Introduction to algorithms.Instructor: Prof. Erik Demaine, Dr. Jason Ku, Prof. Justin Solomon URL: Introduction to Algorithms | Electrical Engineering and Computer Science | MIT OpenCourseWare

COURSE OUTCOME:

Outcome	At the end of the course, the learner will be able to	Bloom's Level	Expected proficiency percentage	Expected Attainment percentage
CO-1	Understand the fundamental concepts of various data structures and their usage.	**3**	**70%**	**65%**
CO-2	Solve problems using data structures such as linear lists, stacks, queues, trees, and graphs and write programs for these solutions.	**2**	**70%**	**60%**
CO-3	Analyze and implement various searching and sorting techniques	**3**	**70%**	**70%**
CO-4	Implement linear and non-linear data structures.	**3**	**70%**	**60%**
CO-5	Apply efficient data structures to solve specific problems.	**4**	**60%**	**50%**

MAPPING OF COURSE OUTCOME TO PROGRAM OUTCOME (CO –PO MAPPING) [Course articulation matrix]

COS/ POS	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	0	2	0	0	0	0	0	0	0	3
CO2	3	3	3	0	3	0	0	0	0	3	0	3
CO3	3	3	0	2	3	0	0	0	0	3	0	3
CO4	3	3	3	0	3	0	0	0	0	3	0	3
CO5	3	3	3	0	3	0	0	0	0	3	0	3

Course utilization Plan / Lesson Plan

Module No	Module Name	Required contact Hours	COs addressed	References book Used
Module-1	**Stack and Queue:**	**10**		
	Introduction to Data Structures, Abstract Data Types (ADT) with examples.	1	1	1-5
	Types of data structures: Primitive, Non-primitive, Categories of Data Structures: Linear vs. Non-linear, Static vs Dynamic.	1	1	1-5
	Stack: Fundamentals of the stack, Representation using an array, and different stack operations.	1	1,2	1-5
	Applications of the Stack: Recursion, Expression conversions (Infix to postfix)	1	1,2	1-5
	Applications of Stack: Expression conversions (Infix to prefix)	1	1,2	1-5
	Applications of Stack: Postfix and prefix expression evaluation	1	1,2	1-5
	Queue: Fundamentals of the queue, Representation of queue using the array	1	1,2	1-5
	Different operations on linear queue and its limitations	1	1,2	1-5
	Circular queue and its implementation using the array	1	1,2	1-5
	Double-ended queue. Different applications of the queue.	1	1,2	1-5
Module-2	**Linked List:**	**8**		
	Array vs Linked list, types of linked lists, Concepts of structure, pointer, and dynamic memory allocation to create nodes of linked list	1	2,3	1-5
	Single linked list: concepts, operations, and implementations	2	2,3	1-5
	Double-linked list: concepts, operations, and implementations	2	2,3	1-5
	Circular linked list; concepts and operations	1	2,3	1-5
	Implementation of stack and queue using linked list	1	2,3	1-5
	Polynomial manipulation using linked list	1	2,3	1-5
Module-3	**Tree:**	**8**		
	Basic terminology, Representation using array and linked list, tree terminologies	1	2,3,4	1-5
	Tree Traversals: Recursive and Non- recursive	1	2,3,4	1-5
	Operations on the binary tree: Finding Height, Leaf nodes, Counting number nodes	1	2,3,4	1-5
	Construction of binary tree from traversal paths	1	2,3,4	1-5
	Binary Search Trees (BST): Insertion, Deletion of nodes to/from BST	1	2,3,4	1-5
	Height Balanced Tree (AVL): Insertion, Rotations on AVL tree, Searching, Deletion,	2	2,3,4	1-5

	Heap, max heap and min heap construction	1		
Module-4	**Graph:**	**8**	2,3,4	1-5
	Graphs: Graph terminologies and their applications	1	2,3,4	1-5
	Representation of graphs (Adjacency Matrix, Adjacency List)	1	2,3,4	1-5
	Graph: Path matrix	1	2,3,4	1-5
	Graph Traversals: Breadth First Search (BFS) and Depth First Search (DFS)	3	2,3,4	1-5
	Spanning Trees, Minimum Spanning Trees	1	2,3,4	1-5
	Shortest Paths: Bellman Ford's Algorithm (Single source)	1	2,3,4	1-5
Module-5	**Sorting and Searching:**	**8**		
	Sorting techniques: Bubble Sort, selection sort, Insertion Sort, Counting sort	2	2,3,4,5	1-5
	Heap Sort	1	2,3,4,5	1-5
	Searching techniques: Linear and Binary search methods (recursive and non-recursive)	2	2,3,4,5	1-5
	Hashing technique: Different types of hash functions	1	2,3,4,5	1-5
	Collision and Collision resolution techniques, Double hashing	2	2,3,4,5	1-5
	TOTAL	**42 HOURS**		

Learning Assessments

Bloom's Level of Cognitive Task		Teacher Assessment / Formative Assessment (40 %)				Summative Assessment (60 %)	
		Quiz (10%)	Assignment (10%)	Experiential learning * (10%)	Attendance (10%)	Mid Sem (20%)	End Sem (40%)
Level-1	Remember	40%	40%	20%		40%	30%
	understand						
Level-2	Apply	60%	40%	40%		60%	50%
	Analyze						
Level-3	Evaluate		20%	40%			20%
	Create						
Total		100%	100%	100%		100%	100%

Department of Computer Science & Engineering	
SUBJECT: Data Structure Lab	
Course Code	CS172
Teaching Hours/Week(L-P-T-E)	(0-3-0-0)
Credits	02
Course offering department	Department of Computer Science and Engineering

Total Contact Hours	30 Hours

Course objectives:

- The Data Structure labs are intended to provide the students a practical knowledge of data structures so that the abstractions discussed in theory classes are translated to concrete solutions.
- It is designed to develop skills among students to understand and implement linear and non-linear data structure and their appropriate usage.
- For the given data structures, the course focus following three aspects - conceptualization, explanation and implementation using the C programming language as an implementation medium.

List of Experiments	
Experiment 1:	Programs to do Stack operation.
Experiment 2:	Programs to do operations on Queue.
Experiment 3:	Programs to do operations on Circular Queue, and Dequeue.
Experiment 4:	Conversions of Infix to Postfix operations and Evaluation of Postfix operations
Experiment 5:	Experiments to do the following operations in a Linked List: i) Creation ii) Insertion iii) Deletion iv) Traversal
Experiment 6:	Experiments to do the following operations in a double Linked List and Circular Linked List: i) Creation ii) Insertion iii) Deletion iv) Traversal
Experiment 7:	Experiments on the Creation of a Tree, Tree Traversal
Experiment8:	Experiments on the Creation and Traversal of a Binary Search Tree, and AVL tree
Experiment 9:	Programs to implement Breadth First Search and Depth First Search in a Graph
Experiment10:	Program to implement PRIMS, Kruskal, and Dijkstra algorithm
Experiment11:	Experiments on Binary search, Bubble Sort, Selection sort, and Insertion Sort
Experiment12:	Program to implement Merge sort, Heap Sort, and Quick Sort

Reference Books:

1. Welcome to Virtual Labs - A MHRD Govt of india Initiative (vlabs.ac.in)
2. Virtual Lab on Programming & Data Structure (iitkgp.ac.in)

COURSE OUTCOME:

Outcome	At the end of the course, the learner will be able to	Bloom's Level	Expected proficiency percentage	Expected Attainment percentage
CO - 1	Develop a strong understanding of stacks, queues and the operations on different forms of linked lists, trees, and graphs, including their operations and use cases.	**3**	**70%**	**65%**
CO - 2	Construct program to perform different operations on a linked list, circular list and double linked list	**2**	**70%**	**60%**
CO - 3	Implement the tree data structure to perform several operations on the same.	**3**	**70%**	**70%**

CO - 4	Develop program to represent graphs by doing carious applications on the graph data structure	**3**	**70%**	**60%**
CO – 5	Implement sorting and searching algorithms, understanding their mechanics and selecting the right approach for specific problems.	**4**	**60%**	**50%**

MAPPING OF COURSE OUTCOME TO PROGRAM OUTCOME (CO –PO MAPPING) [Course articulation matrix]

COS/ POS	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	0	0	3	0	0	0	0	0	0	3
CO2	3	0	0	3	0	0	0	0	0	0	0	3
CO3	3	0	0	3	3	0	0	1	0	0	0	3
CO4	3	3	0	0	3	0	1	0	0	0	0	3
CO5	3	3	0	3	0	1	0	0	0	0	0	3

Course Utilization Plan/ Lesson Plan

NO	Aim of the experiment	Required contact Hours	CO - addressed	Reference used
1	Programs to do Stack operation.	**3**	**1,2**	**1-2**
2	Programs to do operations on Queue.	**3**	**1,2**	**1-2**
3	Programs to do operations on Circular Queue, and Dequeue.	**3**	**1,2**	**1-2**
4	Conversions of Infix to Postfix operations and Evaluation of Postfix operations	**3**	**1,2**	**1-2**
5	Experiments to do the following operations in a Linked List: i) Creation ii) Insertion iii) Deletion iv) Traversal	**3**	**2,3**	**1-2**
6	Experiments to do the following operations in a double Linked List and Circular Linked List: i) Creation ii) Insertion iii) Deletion iv) Traversal	**3**	**2,3**	**1-2**
7	Experiments on the Creation of a Tree, Tree Traversal	**3**	**1, 2, 3**	**1-2**
8	Experiments on the Creation and Traversal of a Binary Search Tree, and AVL tree	**3**	**1, 2, 3, 5**	**1-2**
9	Programs to implement Breadth First Search and Depth First Search in a Graph	**3**	**1, 2, 5**	**1-2**
10	Program to implement PRIMS, Kruskal, and Dijkstra algorithm	**3**	**4,5**	**1-2**
11	Experiments on Binary search, Bubble Sort, Selection sort, and Insertion Sort	**3**	**1, 5**	**1-2**
12	Program to implement Merge sort, Heap Sort, and Quick Sort	**3**	**1, 3, 5**	**1-2**

Learning Assessments

Bloom's Level of Cognitive Task	Continuous Learning Assessment / Formative Assessment (100 %)				
	Record / Observatio	Procedure (10 %)	Lab Experiments	Attenda nce	Viva Voce / Implementati

		n Note (10%)		(50%)	(10%)	on (20%)
Level-1	Remember	40%	40%	20%		40%
	Understand					
Level-2	Apply	60%	40%	40%		60%
	Analyze					
Level-3	Evaluate		20%	40%		
	Create					
Total		100%	100%	100%		100%