Semester 3 - 1

a. **Course Name:** Design of Data Structure

b. **Course Code:** 303105201

c. **Prerequisite:** Computer Programming and Basic Syntaxes

d. **Rationale:** Data structure is a subject of primary importance in Information and Communication Technology. Organizing or structuring data is important for implementation of efficient algorithms and program development. Efficient problem solving needs the application of appropriate data structure during program development.

e. **Course Learning Objectives:**

CLOBJ 1	Gain familiarity with Principles of OSS, Open-Source Standards, Requirements for Software, OSS success, Free Software, Examples, Licensing, Free Vs. Proprietary Software, Free Software Vs. Open-Source Software, Public Domain.
CLOBJ 2	Acquire Knowledge regarding Open-Source History, Open Source Initiatives, Open Standards Principles, Methodologies, Philosophy, Software freedom, Open-Source Software Development, Licenses, Copyright vs. Copy left, Patents, Zero marginal cost, Income-generation Opportunities, Internationalization
CLOBJ 3	Acquire knowledge of Community and Communication, Contributing to Opensource Projects Introduction to GitHub, interacting with the community on GitHub, Communication and etiquette, testing open-source code, reporting issues, contributing code. Introduction to Wikipedia, contributing to Wikipedia or contributing to any prominent open-source project of student's choice. Open-Source Ethics and Social Impact: Open source vs. closed source, Open-source Government, Ethics of Opensource,
CLOBJ 4	Understand GNU/Linux, Android, Free BSD, Open Solaris. Open-Source Hardware, Virtualization Technologies, Containerization Technologies: Docker, Development tools, IDEs, Debuggers, Programming languages, LAMP, Open-Source Database technologies
CLOBJ 5	Demonstrate apache Web server, BSD, GNU/Linux, Android, Mozilla (Firefox), Wikipedia, Drupal, WordPress, Git, GCC, GDB, GitHub, Open Office, LibreOffice Study

f. Course Learning Outcomes:

CLO 1	Use different types of data structures, operations and algorithms.
CLO 2	Apply searching and sorting operations on files
CLO 3	Use stack, Queue, Lists, Trees and Graphs in problem solving.
CLO 4	Implement all data structures in a high-level language for problem solving.

g. Teaching & Examination Scheme:

Teaching Scheme				Evaluation Scheme					
L	T	P	C	Internal Evaluation			ESE		Total
				MSE	CE	P	Theory	P	
3	0	4	5	20	20	20	60	30	150

L- Lectures; **T-** Tutorial; **P-** Practical; **C-** Credit; **MSE-** Mid-Semester Evaluation; **CE-** Continuous Evaluation; **ESE-** End Semester Examination

h. Course Content:

Sr. No.	Topics	Weightage	Teaching Hours
1	**Introduction:** Data Structures, Classifications (Primitive & Non-Primitive), Data structure Operations, Review of Arrays, Structures, Self-Referential Structures, and Unions. Pointers and Dynamic Memory Allocation Functions. Representation of Linear Arrays in Memory, dynamically allocated arrays. Performance analysis of an algorithm and space and time complexities	10%	6
2	**Stacks, Recursion and Queue:** Stacks: Definition, Stack Operations, Array Representation of Stacks, Stacks using Dynamic Arrays, Stack Applications: Polish notation, Infix to postfix conversion, evaluation of postfix expression. Recursion - Factorial, GCD, Fibonacci Sequence, Tower of Hanoi, Queues: Definition, Array Representation, Queue Operations, Circular Queues, Circular queues using Dynamic arrays, Deque, Priority Queues and its problems	15%	8

3	**Linked Lists:** Definition, Representation of linked lists in Memory, Memory allocation; Garbage Collection. Linked list operations: Traversing, Searching, Insertion, and Deletion. Doubly Linked lists, Circular linked lists, and header linked lists. Linked Stacks and Queues. Applications of Linked lists	10%	5
4	**Searching and Sorting:** Interpolation Search Sorts: Selection Sort, Insertion Sort, Bubble Sort, Quick Sort, Merge Sort, Radix Sort	10%	5
5	**Trees:** Terminology, Binary Trees, Properties of Binary trees, Array and linked Representation of Binary Trees, Binary Tree Traversals - In Order, Post Order, Pre Order; Additional Binary tree operations. Threaded binary trees, Binary Search Trees – Definition, Insertion, Deletion, Traversal, Searching, Application of Trees-Evaluation of Expression.	10%	4
6	**Red Black Trees and AVL Trees:** Introduction-Operations on Red Black Trees, AVL tree Construction, Operations on AVL Trees	15%	8
7	**Hashing:** Hash Table organizations, Hashing Functions, Static and Dynamic Hashing	15%	3
8	**Graphs:** Definitions, Terminologies, Matrix and Adjacency List Representation of Graphs, Elementary Graph operations, Traversal methods: Breadth First Search and Depth First Search.	15%	5

i. Text Book and Reference Book:

1. Fundamentals of Data Structures in C, 2ND EDITION, E.Horowitz, S,.Sahni and Susan Anderson- Freed, Universities Press (TextBook)
2. Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Ed, Mc-Graw Hill, 2014.

j. List of Practicals:

1. Implement Stack and its operations like (creation push pop traverse peek search) using linear data structure
2. Implement Infix to Postfix Expression Conversion using Stack.
3. Implement Postfix evaluation using Stack.
4. Implement Towers of Hanoi using Stack.
5. Implement queue and its operations like enqueue, dequeue, traverse, search.
6. Implement Single Linked lists and its operations(creation insertion deletion traversal search reverse).
7. Implement Double Linked lists and its operations(creation insertion deletion traversal search reverse).

8. Implement binary search and interpolation search.
9. Implement Bubble sort, selection sort, Insertion sort, quick sort ,merge sort.
10. Implement Binary search Tree and its operations (creation, insertion, deletion).
11. Implement Traversals Preorder In-order Post-order on BST.
12. implement Graphs and represent using adjacency list and adjacency matrix and implement basic operations with traversals (BFS and DFS).