FACULTY OF COMPUTER SCIENCE

Master of Computer Application (Sem-II)

In Effect from Academic Year 2020-21

Subject Code : 2CS2010202	Subject Title: DATA STRUCTURE
Pre-requisite :	Any programming language like C, C++

Course Objective:

- To extend proficiency in implementation of Data Types.
- To be able to carry out the study of various Algorithms of Time and Space Complexity.
- To get a good quality understanding of applications of Data Structures.

Teaching Scheme (Hours per week)				Evaluation Scheme (Marks)				
				Theory		Practical		
Lecture	Tutorial	Practical	Credit	University Assessment	Continuous Assessment	University Assessment	Continuous Assessment	Total
3	1	3	7	60	40	30	20	150

Subject Contents			
Sr. No	Topic	Total Hours	Weight (%)
1	**Introduction to Data Structures :** Primitive Data Structures, String Manipulation & Pattern Matching, Storage Representation of Strings ,Text Handling, KWIC Indexing	5	10
2	**Linear Data Structures:** Arrays, Storage Structure for Arrays, Structures & Arrays of Structures, Stack, Applications of Stacks, Queues, Priority Queues, Pointers & Linked Allocation , Linked Linear Lists , Circularly Linked Linear Lists , Doubly Linked, Linear Lists, Applications of Linked Linear Lists	14	30
3	**Nonlinear Data Structures-Tree** Operations on Binary Trees, Storage Representation & Manipulation of Binary Trees, Conversion of General Tree to Binary Trees, Sequential & Other Representation of Trees, Application of Trees - Manipulation of Arithmetic Expression, Multi-linked Structures - Sparse Matrices	9	20
4	**Nonlinear Data Structures-Graph** Matrix Representation of Graphs, Graphic Representation of List Structures , Other Representation of Graphs, Breadth First Search (BFS), Depth First Search (DFS) , Spanning Trees , Garbage Collection	8	15
5	**Sorting & Searching:** Introduction, Selection Sort, Bubble Sort , Merge Sort , Heap Sort , Quick Sort ,Radix Sort , Sequential Searching , Binary Searching , Search Trees – Height Balanced , 2-3 Trees , Weight Balanced , m-ary Trees , Trie Structures , Hash Table Search Methods , Introduction, Hashing Functions , Collision Resolution Techniques	12	25

Course Outcome:

At the end of this course, the student would be able

- To choose the appropriate data type and data structure for a given problem.
- To choose the best algorithm to solve a problem by considering various problem characteristics, such as the data size, the type of operations, etc.
- To create the algorithms and program of various operations on Queues, Stacks, Linked Lists, Trees, Graphs, Sorting, Searching, Hash tables
- To evaluate algorithms with respect to time and space complexity

List of References:

1. "An Introduction to Data Structures with Applications", Jean-Paul Tremblay, Paul G. Sorenson, Tata McGraw-Hill, 2nd Edition, (2007).
2. "Data Structures Via C++: Objects by Evolution", A. Michael Berman, , Oxford Univ. Press (2004)
3. "Fundamentals of Data Structures in C", Horowitz, Sahni, Anderson-Freed, , University Press (2nd edition-2007)
4. "Data Structures Using C & C++", Tenenbaum, PHI.
5. "Sorting & Searching - The Art of Computer Programming" D E Knuth, , Vol. 3, Pearson Education (1998).

List of Experiments:

Note: The experiment list provided beneath is for reference only. The course teacher may Change/formulate it as per his/her methodology and requirement.

Sr. No	Practical Exercise
1.	**Array**
	1.1 Write a program to insert, Traversing, Delete, Sorting a single Dimension Array.
	1.2 Write a program to Marge two array lists.
	1.3 Write a program Insert, traversing, Delete, Sorting a 2-Dimension Array.
	1.4 Write a program for Addition, Subtraction, Multiplication and Transpose of matrices.
2.	**Stack**
	2.1 Implement Stack by using static & dynamic storage representation. (Push, pop, peep, display, isempty, isfull)
	2.2 Write a program to solve the Tower of Hanoi problem. (Using recursion & without Recursion).
	2.3 Write a program to print n Fibonacci series (using recursion).
	2.4 Write a program to find factorial of n number (using recursion).
	2.5 Write a program to find GCD & LCD of two numbers (using recursion). 10. Convert decimal number into binary. (Using recursion).
	2.6 Find X^N, i.e. power (x,n) (using recursion).
	2.7 Write a program to convert infix expression into postfix expression.(Vice versa) 13. Write a program for evaluation of postfix expression.
	2.8 Implement Double Stack, i.e. maintain two stacks in the same shared array.
3.	**Queue**
	3.1 Write a program for Single Queue operation (Insertion, Deletion).
	3.2 Write a program for Circular Queue operations (Insertion, Deletion).
	3.3 Write a program for Double Ended Queue operations (Insertion, Deletion).
	3.4 Write a program for Priority Queue operations (Insertion, Deletion).
4	**Link List**
	(Operations: Create Display, Insert, Delete, Search, Count, Copy, Concat, Reserve, Sort, Merge, Union, Insertion, etc.)
	4.1 Write a menu driven program that implements singly linked list for the above operations.
	4.2 Write a menu driven program that implements doubly linked list for above operations.
	4.3 Write a menu driven program that implements circular linked list for the above operation
5	**Tree & Graph**
	5.1 Write a program to create a binary search tree and print it's elements in order (Write iterative code).
	5.2 Write a program to create a binary search tree and print it's elements in preorder (Write iterative code).

5.3 Write a program to create a binary search tree and print it's elements in postrder (Write iterative code).
5.4 Write a program to delete an element from a binary search tree.
5.5 Write a program to make another copy of a given binary search tree.
5.6 Write a program to create a graph in an adjacency matrix. Generate path matrix from the Given matrix.
5.7 Write a program to create a graph in an adjacency list structure. (Node directory structure) traverse it in DFS.
5.8 Write a program to create a graph in a adjacency list structure. (Node directory structure) traverse it in BFS.

6 Sorting & Searching

6.1 Write a program for linear Search (With Recursion & without Recursion)
6.2 Write a program for binary Search (With Recursion &without Recursion)
6.3 Write a program for implementation of Bubble Sort.
6.4 Write a program for implementation of Insertion Sort.
6.5 Write a program for implementation of Quick Sort.
6.6 Write a program for implementation of Selection Sort.