4145 -- DATA STRUCTURES

Teaching Schedule Per Week			Progressive Assessment		Examination Schedule (Marks)			
					Theory		Practical Ex.	Total
Lectures	Practical	Credit						
3	3	6	25	25	3 Hrs	100	50	200

Pre-requisite	Source
3030	COM

Semester	Theory	Test	Total	TW	PR	Gr Total

Rationale: Computer Science is primarily concern with the study of data structures and their transformation by suitable means. Any algorithm has to operate in Data Structures to generate useful programs. A sound base in this subject is extremely important to understand other advanced topics of computer science. This is one reason why any curriculum in computer software discipline invariably provides at least one course in data structure. The course exposes the student to different data structures, their relative merits, applications and implementations. Sorting and searching techniques are also included. String manipulation is another key topic hat is included in this course.

COURSE CONTENTS	Hrs	Mks
1. BASIC CONCEPTS OF DATA REPRESENTATION Abstracts and system defined data types; Representation, primitive data structures	1	5
2. ALGORITHMIC DESIGN Design and analysis of algorithm; Top-Down and Bottom-Up approaches to algorithms design; Analysis of Algorithms:-Frequency count, Complexity measures in terms of time and space; Structured approach to programming	5	10
3. ARRAYS Representation of arrays: single and multidimensional arrays. Address calculation using columns and rows major ordering.	5	10
4. STACKS AND QUEUES Representations of stacks and queues using arrays; Circular queues Application of stacks; Conversion from infix to postfix and prefix expressions, Evaluation of postfix expression using stacks.	5	10
5. LINKED LISTS Singly linked lists: operations on list, linked stacks and queues, Polynomial representation and manipulation using linked lists, circular linked lists, Doubly linked lists, Generation lists, Sparse matrix representation using generalised list structures.	5	10
6. STORAGE ALLOCATION & GARBAGE COLLECTION Memory allocation strategies - First fit and best fit approach, Boundary tag method; Memory freeing algorithms in each case.	5	10
7. TREES Binary Tree traversal methods - Preorder traversal, Inorder traversal, Post Order traversal, Recursive and non-recursive algorithm for above mentioned traversal methods. Representation of trees and its applications - Binary tree representation of a tree, Conversion of forest into tree. Threaded binary tree; Decision and game tree.	5	10
8. SYMBOL TABLES Static symbol table, Hash tables, Binary search tree, Dynamic tree tables - Height balanced (AVL) Tree, B - Trees	5	10
9. SEARCHING AND SORTING Searching - Sequential search, Binary search, Fibonacci search, Indexed search, Hashing scheme. Internal Sorting - Insertion sort, Selection Sort, Bubble sort, Quick sort, Merge sort, Heap sort, Sorting on multiple keys	2	5
10. STRINGS AND THEIR FEATURES String representation, String manipulation using arrays and lists, String matching algorithms.	5	10
11. GRAPHS Graph representation - Adjacency matrix, Adjacency list, Adjacency multilists. Traversal schemes - Depth first search, Breadth first search	5	10
Total	48	100

PRACTICALS:

The practical work will be based on a set of programming exercises on the topics given below:

1. Linear Data Structures – a) Contiguous memory allocation - Processing in an array, Processing in a stack, Processing in a queue, Application of stack. b) Linked Memory Allocation - Processing in a linked list, Processing in a stack, Processing in a queue.
2. Non-Linear Data Structure – a) Tree Structures - Processing in a binary search tree, Processing in a thread binary tree. b) Graph Structure - Depth first search, Breadth first search
3. Sorting Techniques - Quick sort, Heap sort, Merge sort, Insertion sort, Selection sort, Index sort
4. Searching techniques - Sequential search, Binary search, Fibonacci search, Indexed search
5. String Manipulation