

Subject Code: 1ET4000304	Subject Title: Data Structure
Pre-Requisite	C Programming

RATIONALE:

Development of application systems and software that use underlying architecture of machines efficiently and effectively requires the ability to use and manipulate various types of Data Structures and other constructs. This being a fundamental ability which is language neutral, yet requires use of a language for its implementation. This is a basic course which goes along with other programming courses to develop an integrated ability to efficient software development, hence this course is very important for computer and IT diploma engineers.

COURSE OBJECTIVE:

Following are the objectives of this course:

- Understand and remember algorithms and its analysis procedure..
- Introduce the concept of data structures through ADT including List, Stack and Queues.
- To design and implement various data structure algorithms.
- To introduce various techniques for representation of the data in the real world.
- To develop application using data structure algorithms.

TEACHING AND EVALUATION SCHEME:

Teaching Scheme (Hours per week)				Evaluation Scheme (Marks)			
Lecture	Tutorial	Practical	Credit	Theory		Practical	
				University Assessment	Continuous Assessment	University Assessment	Continuous assessment
3	-	2	5	60	40	20	30

DETAILS OF COURSE CONTENTS:

Sr. NO.	TOPIC	Total Hrs.	WEIGHTAGE
1	Basic Concepts of Data Structures 1.1 Data Structure Basic Concepts 1.2 Types of data structures 1.3 Primitive and non-primitive data structures. 1.4 Introduction to Algorithms. 1.5 Key features of an algorithm. 1.6 Analysis Terms (for the definitions purpose only) : a. Time Complexity b. Space Complexity c. Asymptotic Notations ,Big 'O', Notation , Best case Time Complexity, Average case Time Complexity, Worst case Time Complexity	4	10%

	1.7 Array : i. Row Major Arrays ii. Column Major Arrays 1.8 Overview of various array operations. 1.9 Searching an element into an array – i. Linear Search ii. Binary Search		
2	Stack and Queues 2.1 Linear and Non-Linear Data Structures 2.2 Stack : Array representation of Stack, PUSH POP Operations on Stack, Implementation of Stack, Application of Stack, Infix, Prefix and Postfix Forms of Expressions, Recursive Functions (factorial, greatest common divisor, Fibonacci series) 2.3 Queue: Array representation of Queue, Operations on Queue, Implementation of Queue, Limitation of Single Queue 2.4 Concepts of Circular Queue 2.5 Application of queue 2.6 Difference circular queue and simple queue	10	20%
3	Linked List 3.1 Pointers Revision 3.2 Revision of Structure 3.3 Revision of structure using pointers 3.4 Dynamic Memory Allocation 3.5 Linked list Presentation 3.6 Types of Linked List 3.7 Basic operations on singly linked list : Insertion of a new node in the beginning of the list, at the end of the list, after a given node, before a given node, in sorted linked list, Deleting the first and last node from a linked list, Searching a node in Linked List, Count the number of nodes in linked list 3.8 Concepts of circular linked list 3.9 Difference between circular linked list and singly linked list 3.10 Basic operations on Doubly linked list : Insertion of a new node in the beginning of the list, at the end of the list, after a given node, before a given node. Deleting the first and last node from a linked list, Searching a node in Linked List, Count the number of nodes in linked list 3.11 Applications of linked list	12	25%
4	Sorting and Hashing 4.1 Sorting Methods : a. Bubble Sort, b. Selection Sort, c. Quick Sort, d. Insertion Sort, e. Merge Sort,	8	20%

	f. Radix Sort 4.2 Hashing Concepts 4.3 Division Method, Middle Square Method, Folding Method 4.4 Collision in Hashing 4.5 Collision resolution techniques: Linear Probing		
5	Trees & Graphs 5.1 Non-linear data structures 5.2 Binary trees : Complete Binary Tree, Basic Terms: level number, degree, in-degree and out-degree, leaf node, similar binary trees, copies of binary trees, directed edge, path, depth, General Tree, Conversion of General Tree to Binary Tree, Forest. 5.3 Binary Search Tree : Insertion of a node in binary tree, Deletion of a node in binary tree, Searching a node in binary tree 5.4 Tree Traversal : Inorder, Preorder, Postorder 5.6 Applications of binary tree 5.7 Graph: Representation Of Graphs 5.8 Elementary Graph operations,(Breadth First Search, Depth First Search, Spanning Trees, Shortest path, Minimal spanning tree)	10	25%

COURSE OUTCOME:

After revising this course, student will be able to:

1. Select appropriate data structures as applied to specified problem definition.
2. Implement operations like searching, insertion, and deletion, traversing mechanism etc. on various data structures.
3. Students will be able to implement Linear and Non-Linear data structures.
4. Implement appropriate sorting/searching technique for given problem.
5. Design advance data structure using Non-Linear data structure.
6. Determine and analyze the complexity of given Algorithms.

LIST OF PRACTICAL:

SR. No.	Experiment
1	Define various terms such as algorithm, various approaches to design an algorithm, time complexity, space complexity, big 'o' notation, best case, average case and worst case time complexity etc.
2	Implement Sequential search algorithms
3	Implement Binary search algorithms
4	Implement push and pop algorithms of stack using array
5	Implement recursive functions

6	Implement insert, delete algorithms of queue using array
7	Implement insert, delete algorithms of circular queue
8	Implement insertion of node in the beginning of the list and at the end of list in singly linked list
9	Implement insertion of node in sorted linked list
10	Implement insertion of node at any position in linked list
11	Implement searching of a node, delete a node and counting no of node algorithms in singly linked list
12	Implement insertion of node in the beginning and at the end of doubly linked list
13	Implement insertion of node at any position in doubly linked list
14	Implement searching of a node, delete a node and counting no of node algorithms in doubly linked list
15	Implement Bubble sort, Selection sort algorithms
16	Implement Quick Sort
17	Implement Insertion sort
18	Implement Merge Sort algorithm
19	Solve hash table example using division method, method square method, folding method (paper work only)
20	Implement construction of binary search tree
21	Implement inorder, preorder and postorder traversal methods in binary search tree
22	Implement searching algorithm in binary search tree

LIST OF REFERENCES:

Sr.no.	Title	Author
1	Data and File Structures using C	Thareja, Reema
2	Data Structures using C	ISRD Group
3	Data Structures	Chitra, A Rajan, P T
4	Classic Data Structures	Samanta, D.