D. Syllabus Detailing and Learning objectives

Module	Chapter	Detailed Content	Syllabus Detailing	Learning Objectives
Module1	**Introduction to Data structures and Analysis (06 Hours)**	Introduction to Data structures,Need of Data structures,Types of Data structures : Linear and non linear data structures Arrays, Stacks,Queue,Linked listand Tree,Graph,Recursion, ADT (Abstract Data type). Introduction toAnalysis,Algorithms, characteristics of an algorithms,Time and Space complexities, Order of growth functions, Asymptotic notations	**Purpose**: To know various ADT's(Abstract Data Types), and to do analysis of an algorithm with its use in real time applications along with the implementations **Scope –** **1. Academic Aspects-** Understanding various Data structures , characteristics of an efficient algorithm and ways to represent algorithm complexity. **2. Technology Aspect- Programing u**sing C **3. Application Aspect-** to find suitable data structure based on application need. Students Evaluation – Theory Questions to be asked on 1. What is ADT. 2. List various ADT's 3.Llist characteristics of efficient Algorithm 4. Explain various asymptotic notations used to represent complexity of an algorithm	1. To **describe** various ADTS.**(U)** 2. To **Distinguish** linear and nonlinear data structures.(**AN**) 3. To determine complexity of an algorithm**.(AN)**
Module 2	**Stack (05Hours)**	Introduction to Stack, Stack as ADT, Operations on stack, Application of stack: – reversing string, Polish	**Purpose-** To understand Stack as ADT. To implement stack operations and its applications like recursion, postponement data usage etc. Scope – 1. **Academic Aspects**- understanding use of stack in various applications.	1.**Define** Stack.**(R)** **2**. **Explain** operations and applications of stack (R) 3. **Interpret** stack to covert infix expression to postfix expression**(A)**

		notations (PCB), Threads, Thread management. Process Scheduling: Types, Comparison of different scheduling policies.	2. **Technology Aspect**- Using C Programming Language 3. **Application Aspect**- Developing data postponement usage using stack **Students Evaluation –** 1. Define Stack 2. List Operations of Stack 3. Discuss Applications of Stack? 4.Implement Stack and its applications using static and dynamic memory allocation techniques	
Module 3	**Queue (06 Hr)**	Introduction to Queue,Queue as ADT, Operations on Queue, Linear representation of queue, Circular Queue, Priority Queue, De-queue, Application of Queues	**Purpose** – To understand Queue as ADT. To implement Queue operations and its applications like processes scheduling algorithms **Scope** – **1. Academic Aspects-** understanding use of Queue in various applications. 2**. Technology Aspect**- Using C Programming Language **3. Application Aspect-** Developing Process Scheduling algorithms using Queue **Students Evaluation –** 1. Define Queue 2. List Operations of Queue 3. Discuss Applications of Queue? 4.Implement Queue and its applications using static and dynamic memory allocation techniques	1.**Define** Queue.**(R)** **2**. **Explain** operations and applications of Queue (R) 3. **Interpret** Queue to implement various types of scheduling algorithms**(A)**

Module 4	**Linked List (8 Hours)**	Introduction to Linked List, Basic concept of Linked List, Memory allocation&de allocation of Linked list, Singly Linked list, Doubly Linked list, Circular linked list, Operations on linked list,Linked representation of stack,Linked representation of Queue, Application of linked list.	**Purpose-** This chapter is focused on dynamic memory management techniques by implementing various ADT's like List, Stack and Queue. **Scope –** 1**. Academic Aspects**- understanding Memory management techniques 2**. Technology Aspect**- Using C Programming Language 3. **Application Aspect**- To implement applications like polynomial representations using linked list . **Student Evaluation -** **Student Evaluation -** 1. Describe malloc and calloc 2. What is Linked List (Single, Double and Circular)? 3. Compare Static vs dynamic memory allocation. 4. Explain Applications of Linked List	1. D**efine** malloc and calloc **(R)** **2. Define** types of linked list **(U)** **3. Explain Compaction (U)** 4**. Compare** Static and Dynamic memory allocation**(E)** **5. interpret** linked list to represent polynomials**(A)**
Module 5	**Sorting and Searching (12 Hours)**	Introduction to Sorting: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort,	**Purpose –** To apply sorting technique on real world data sets and re-arrange in ascending or descending order so that it is easier and faster to locate items in a sorted list.	**Learning Objective:** 1. Develop Algorithm and Implement

		Heap Sort, Shell Sort, Radix sort. Analysis of Sorting Techniques. Comparison of sorting Techniques Introduction to Searching: Linear search, Binary search, Hashing Techniques, Different Hash functions, Collision& Collision resolution techniques, Analysis of searching Techniques	**Scope –** **1. Academic Aspects-** To learn internal and external sorting techniques and also different searching methods. **2. Technology Aspect-** Many Programming Languages provides API for Sorting and searching the data. **3. Application Aspect-** **i)** Merge Sort: Databases use an external merge sort to sort sets of data that are too large to be loaded entirely into memory. The driving factor in this sort is the reduction in the number of disk I/Os. ii) Bubble Sort: Bubble sort is used in programming TV remote to sort channels on the basis of longer viewing time. iii) Heap Sort: Heap sort is used in reading barcodes on plastic cards. The service allows to communicate with the database to constantly run checks to ensure that they were all still online and had to constantly report statistics on which readers were performing the worst, which ones got the most/least user activity, etc. **iv)Quick Sort:** Sports scores are organized by quick sort on the basis of win-loss ratio. **v) Radix Sort:** eBay allows you to sort listings by the current Bid amount leveraging radix sort. **vi) Selection Sort:** K12 education portal allows sorting list of pupils alphabetically through selection sort.	operations like sorting, searching mechanism etc. on various data structures. 2. Analysis of Sorting techniques 3. Analysis of searching techniques 4. Comparison of all sorting techniques

			Student Evaluation – 1.What do you mean by internal and external sorting techniques? 2. Explain algorithm, analysis and code for i)Merge sort ii)Quick sort iii)Radix Sort iv) Insertion sort v) Shell sort. 3. Compare all sorting techniques	
Module 6	**Trees &Graph (10 Hours)**	Introduction to Trees, Definitions& Tree terminologies, Binary tree representation, Operations on binary tree, Traversal of binary trees, Binary search tree, Threaded Binary tree, Expression tree, Application of Trees Introduction to Graph, Introduction Graph Terminologies, Graph Representation, Type of graphs, Graph traversal: Depth first	**Purpose –** To **understand how** tree (nonlinear data structure) for organizing data objects in terms of hierarchical relationships **Scope –** **1. Academic Aspects-** Student are learning tree basics, types of tree, Designing of algorithm for building BST, tree traversal techniques, AVL tree, graph applications, graph basics, types of graph, representation of graph, graph Traversal Techniques and Minimum cost spanning tree etc. **2. Technology Aspects:** Implementation of Trees concepts is used in File system, **Design** FAT and INODES, data mining and pattern recognition etc. Graph is useful in networking, geographical information system, ecommerce etc. **3 Applications:** **Tree applications:**	**Learning Objective:** • **Define tree** • Describe the working and implementation of tree. • To describe how trees are used to implement the file system of several popular operating systems. • To explain how trees can be used to evaluate arithmetic expressions. • To understand how to use trees to support searching operations in O(log n) average time, and how to refine these ideas to obtain O(log n) worst-case bounds. We will also see how to implement these operations when the data is stored on a disk. • To define and explain the basics of graph • To understand the how to store graph into computer's memory. • To learn shortest path algorithms

			1. Application Aspect- Decision Making: Next Move in games computer chess games build a huge tree (training) which they prune at runtime using heuristics to reach an optimal move. **2. Networking:** Router algorithms -Network Routing, where the next path/route of the packet is determined. Social networking is the current buzzword in CS research. It goes without saying that connections/relations are very naturally modeled using graphs. Often, trees are used to represent/identify more interesting phenomena. **3.Representation** Chemical formulas representation XML/Markup parsers use trees. Producers/consumers often use a balanced tree implementation to store a document in memory. **4. Manipulate Hierarchical** Data Make information easy to search Manipulate sorted lists of data. **5. Workflow** As a workflow for compositing digital images for visual effects. **Graph Application** 1. Graph theory is also used to study molecules in chemistry and Physics. 2. Recommendation on e-commerce websites: "The Recommendation for you" section on various	• To classify and compute minimum cost spanning trees and explain its applications • To describe graph traversal techniques

			ecommerce websites uses graph theory to recommend items of similar type to user's choice. 3. Google Maps: Various locations are represented as vertices and roads are represented as edges and graph is used to find shortest path between two nodes	
			Student Evaluation **Tree:** 1.**Explain** Search and Insertion in BST 2.**Explain** the deletion from BST 3.**Compute** Minimum value in a Binary Search Tree 4.**Find** Inorder predecessor and successor for a given key in BST 5.Check if a binary tree is BST or not 6.**Find** Lowest Common Ancestor in a Binary Search Tree. 7.**Find** norder Successor in Binary Search Tree 8.**Find** k-th smallest element in BST 9.**Merge** two BSTs with limited extra space 10.**How** two nodes of a BST are swapped, correct the BST 11.**Find** a pair with given sum in a Balanced BST 12.**Compute** Total number of possible Binary Search Trees with n keys 13.**How to Merge** Two Balanced Binary Search Trees 14.**Conversion** from Binary Tree to Binary Search Tree 15.**Explain** Binary Tree Properties 16.**Types** of Binary Tree	

			17.**Applications** and **uses** of tree data structure 18.**Describe** Tree Traversals technique 19.**Compare** BFS vs DFS for Binary Tree 20.**Explain** Threaded Binary Tree 21.**Construct** Tree from given In order and Preorder traversals **Graph:** 1.Develop the algorithm and code for finding shortest Path algorithm using dijiktra's Algorithm 2. Develop the algorithm for Minimum cost spanning trees using Prims' and Kruskal Algorithm 3. Develop algorithm and code for following graph traversal techniques such as BFS and DFS	