**Data Structures and its Application – UE22CS252A (5-0-2-5-5)**

**# of Sessions: 112**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Class #** | **Unit #**  **Chapter Title/Reference Literature** | **Topics to be Covered** | | | **% of Portion covered** | | |
|  | | | | **% of Syllabus** | | **Cumulative %** | |
| 1 | Unit 1: Linked List and Stack  T1/R1 | | Overview of the C, Introduction to Data Structures. | | **25** | | **25** |
| 2 | Pointers, Structures | |
| 3 | Functions & Recursion | |
| 4 | Static and Dynamic Memory Allocation | |
| 5 | Abstract Data Type (ADT), List as an ADT, List as a data structure | |
| 6 | List Implementation using Array. | |
| **7,8** | **LAB-1** | |
| 9 | Singly Linked List (SLL) insert operations: beginning, end | |
| 10 | SLL : insert & delete at a specified position, destroy list operation | |
| 11 | SLL delete operations: beginning, end, at a specified position | |
| 12 | SLL Operations : other operations like search operation, concatenate etc.. | |
| 13 | Doubly Linked List (DLL) insert operations: beginning, end, at a specified position, destroy list operation | |
| 14 | DLL delete operations: beginning, end, at a specified position, search operation | |
| **15,16** | **LAB 2** | |
| 17 | Ordered Double Linked List | |
| 18 | Circular Singly Linked List | |
| 19 | Circular Doubly Linked List | |
| 20 | Sparse matrix and its representation using Multi list, Case Study – Skip List | |
| 21 | Basic structure of a stack, stack using arrays | |  |
| 22 | Stack using linked list. | |
| **23,24** | **LAB 3** | |
| 25 | Applications of stack: Function execution, Nested functions. | |
| 26 | Applications of stack: Conversion of an expression from Infix to postfix / prefix | |
| 27 | Applications of stack: Evaluation of a postfix expression using stack | |
| 28 | Applications of stack:Parenthesis matching using stack | |
| 29 | Unit 2 – Queue & Binary Trees  T1/R1 | | Basics of Queue Data Structure, Linear Queue using Array. | | **25** | | **50** |
| 30 | Queue using Linked List | |
| **31,32** | **LAB-4** | |
| 33 | Circular queue using array and linked list | |
| 34 | Priority queue using array | |
| 35 | Priority queue using linked list | |
| 36 | Double ended queue (Deque) using array and linked list | |
| 37 | Other Queue Operations : Stacks usig Queue, Queue using Stacks | |
| 38 | Applications of Queue: Case Study – Josephus problem | |
| **39,40** | **LAB 5** | |
| 41 | Binary Tree and Binary Search Tree (BST) : definition, properties | |
| 42,43 | Binary Trees using Dynamic Memory Allocation. | |
| 44 | n-ary tree, Forest, conversion of an n-ary tree and Forest to Binary Tree | |
| 45 | BST using arrays | |
| 46 | BST using Linked List | |
| **47,48** | **LAB 6** | |
| 49 | BST Traversal: Inorder, preorder, postorder. | |
| 50 | Other tree operations-finding the height, depth, count no of nodes, leaf nodes | |
| 51,52 | Node deletion operation on a Binary Search Tree | |
| 53,54 | Binary Search Tree Traversals using iteration. | |
| **56,57** | **LAB-7** | |
| 58,59 | Revision / HALP | |
| 60,61 | Unit 3 – Application of Trees , Basics of Graphs  T1/R1 | | Expression Tree | | **25** | | **75** |
| 62,63 | Threaded Binary Tree | |
| 64 | Heap Tree and its properties | |
| **65,66** | **LAB-8** | |
| 67 | Heap using Array - Bottom up Heap construction | |
| 68,69 | Heap using Array – Top dowm Heap Construction | |
| 70 | Priority Queue using min and max heap | |
| 71 | Balanced Tree : Definiation, AVL Tree | |
| **72,73** | **LAB 9** | |
| 74,75 | Rotation in AVL Tree | |
| 76 | Splay Tree | |
| 77 | Graphs: Introduction, properties. | |
| 78 | Graphs : Types og Graphs, Applications and Representation | |
| 79 | Graphs using adjacency matrix and Adjacency List | |
| **80,81** | **ASSIGNMENT / Mini Project** | |
| 82 | Graph Traversal Techniques – DFS & BFS with examples | |
| 83 | Depth First Search (DFS) traversal of a graph | |
| 84 | Breadth first search (BFS) traversal of a graph | |
| 85 | HALP/Revision | |
| 87 | Unit 4: Applications of Graph, Hashing, Trie ,Suffix Trees  T1/R1 | | Application of BFS and DFS: Connectivity of graph | | **25** | | **100** |
| 88 | Application of BFS and DFS: finding path in a network | |
| 89 | Application of BFS and DFS: To check if there exists a cycle in a given graph. | |
| **90,91** | **LAB 10** | |
| 92,93 | Case Study: Indexing in databases (B Tree: K-way tree), Introduction, Properties | |
| 94,95 | Hashing: Simple mapping, hash function, hash table | |
| 96 | Collision handling using seperate Chaining. | |
| **97,98** | **LAB 11** | |
| 99,100 | Collision handling using linear and quadratic probing. | |
| 101,102 | Collision handling using double hashing and rehashing. | |
| 103,104 | Introduction to Trie trees, properties, Application. | |
| **105,106** | **LAB-12** | |
| 107 | Trie trees: Insert & Search, Delete | |
| 108 | Application of Trie : Display the words in a trie in lexicographic order, Prefix based search, multi pattern search | |
| 109 | Application of Trie : Word Prediction, Auto Complete Feature | |
| 110 | Suffix Tree : Introduction, Properties, Construction. | |
| 111 | **Mini Project/ Assignment** | |
| 112 | Revision/ HALP | |

**Tool/ Languages:** C Programming Language

**Text Book:**

1. “Data Structures using C / C++” , Langsum Yedidyah, Moshe J Augenstein, Aaron M Tenenbaum Pearson Education Inc, 2nd edition, 2015.

**Reference Book:**

1. “Data Structures and Program Design in C”, Robert Kruse, Bruce Leung, C.L Tondo, Shashi Mogalla, Pearson, 2nd Edition, 2019.

**Evaluation Policy**

|  |  |  |
| --- | --- | --- |
|  | **Conducted** | **Scaled to** |
| ISA -1 (Unit 1 & Unit 2) | 40 | 20 |
| ISA -2 (Unit 3 & Unit 4) | 40 | 20 |
| Experiential Learning / HALP (Assignment +Mini Project) | 20 | 10 |
| Lab (Total 10 Labs) | 10 | 20 |
| **ESA** | **100** | **50** |
| **Total** |  | **120(scaled to 100)** |