**FR. Conceicao Rodrigues College Of Engineering**

Father Agnel Ashram, Bandstand, Bandra-west, Mumbai-50

**Department of Information Technology**

**S.E. (IT) (semester III)  (2019-2020)**

**Lesson Plan**

**Subject: Data Structures and Analysis (ITC303)**

**Credits-4**

SYLLABUS

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.****No.** | **Module**  | **Detailed Content**  | **CO****Mapping** |
| 00 | Prerequisite | C Programming Language |  |
| 01 | IntroductiontoData structures andAnalysis | Introduction to Data structures,Need of Data structures, Types of Data structures : Linear and nonlinear data structuresArrays, Stacks, Queue, Linked listand Tree, Graph, Recursion, ADT(Abstract Data type).Introduction to Analysis,Algorithms, characteristics of analgorithms, Time and Spacecomplexities, Order of growthfunctions, Asymptotic notations | CO1 CO2CO3CO6 |
| 02 | Stack | Introduction to Stack, Stack asADT, Operations on stack,Application of stack: – reversingstring, Polish notations | CO1 CO2CO3CO6 |
| 03 | Queue | Introduction to Queue, Queue asADT, Operations on Queue, Linearrepresentation of queue, CircularQueue, Priority Queue, De-queue,Application of Queues | CO1 CO2CO3CO6 |
| 04 | Linked list | Introduction to Linked List, Basicconcept of Linked List, Memoryallocation & de allocation ofLinked list, Singly Linked list,Doubly Linked list, Circular linkedlist, Operations on linked list,Linked representation of stack,Linked representation of Queue,Application of linked list. | CO1 CO2CO3CO6 |
| 05 | Sorting andSearching | Introduction to Sorting: BubbleSort, Selection Sort, Insertion Sort,Quick Sort, Merge Sort, Heap Sort,Shell Sort, Radix sort. Analysis ofSorting Techniques. Comparison ofsorting TechniquesIntroduction to Searching: Linear search, Binary search, HashingTechniques, Different Hashfunctions, Collision& Collisionresolution techniques, Analysis ofSearching Techniques. | CO4CO5CO6 |
| 06 | Trees & Graph | Introduction to Trees, Definitions&Tree terminologies, Binary treerepresentation, Operations onbinary tree, Traversal of binarytrees, Binary search tree, ThreadedBinary tree, Expression tree,Application of TreesIntroduction to Graph, IntroductionGraph Terminologies, GraphRepresentation, Type of graphs,Graph traversal:Depth firstsearch(DFS)&Breadth Firstsearch(BFS), Minimum SpanningTree : Prim’s &Kruskal’s ShortestPath Algorithm – Dijkstra’sAlgorithm. Applications of graph | CO1CO2 CO3 CO6 |

Internal Assessment:

Consisting of Two Compulsory Class Tests Approximately 40% to 50% of syllabus content must be covered in First test and remaining 40% to 50% of syllabus contents must be covered in second test.

**CO-Statements:**

CO1 Select appropriate data structure as applied to specified problem definition.

CO2 Implement operations like searching, insertion, deletion, traversing mechanism etc. on various data structures.

CO3 Implement linear and non-linear data structures.

CO4 Implement appropriate sorting/searching technique for given problem.

CO5 Design advance data structure using Non-Linear data structure

CO6 Determine and analyse the complexity of given algorithms.

**CO-PO-PSO Mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course Name** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** |
| CO1 | 1 | 2 |  |  |  |  |  |  |  |  |  |  |  |  |
| CO2 |  | 2 | 2 |  |  |  |  |  |  |  |  |  | 1 |  |
| CO3 |  | 2 | 2 |  |  |  |  |  |  |  |  |  |  1 |  |
| CO4 |  |  2 |  2 |  |  |  |  |  |  |  |  |  |  1  |  |
| CO5 |  |  2 |  2 |  |  |  |  |  |  |  |  |  |  |  |
| CO6 |  |  2 |  |  1 |  |  |  |  |  |  |  |  |  |  |

**CO Assessment Tools**

|  |  |  |
| --- | --- | --- |
|  | **Direct Methods** | **Indirect Methods** |
|  | **Test1** | **Assig1** | **Lab Work** | **Test2** | **Assig2** | **University Theory Result** | **University Theory Result** | Course Exit Survey |
| CO1 | **30%** | **20%** |  |  |  | **25%** | 25% | 100% |
| CO2 | **25%** | **25%** |  |  |  | **25%** | 25% | 100% |
| CO3 | **20%** | **20%** |  | 20% |  | **20%** | 20% | 100% |
| CO4 |  |  | **30%** |  | **30%** | **20%** | 20% | 100% |
| CO5 |  |  | **30%** | **30%** |  | **20%** | 20% | 100% |
| CO6 |  |  |  | **30%** | **30%** | **20%** | 20% | 100% |

**Curriculum Gap/**

**Content beyond syllabus**

**Lecture Plan:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No of classes available:** | **44** | **No of Classes taken:** | **43** |
| **Sr. No.** | **Topic Planned** | **Planned Date**  | **Actual Date** | **Delivery Mechanisms** |
| **1** | **Introduction to Data Structures, Definition, Types of Data Structures** | **3/7** | **3/7** | **Chalk and Black Bard** |
| **2** | **Linear Data Structures: Array, Stacks, Queue, Linked List** | **4/7** | **4/7** | **Chalk and Black Bard** |
| **3** | **Non-Linear Data Structures: Tree Graph****Recursion** | **5/7** | **5/7** | **Chalk and Black Bard** |
| **4** | **ADT: what is ADT? Example of Fraction ADT.**  | **10/7** | **9/7** | **Chalk and Black Bard** |
| **5** | **Stack: Introduction to stack, Stack as ADT, operations on stack, stack menu driven program** | **11/7** | **10/7** | **Chalk and Black Bard** |
| **6** | **Application of stack:** 1. **Reversing String using stack: examples and algorithm**
2. **Expression validating: examples**
 | **12/7** | **11/7** | **Chalk and Black Bard** |
| **7.**  | **Application of stack continued….:** **Expression validating: algorithm**1. **Infix to postfix: examples on infix to postfix**
 | **15/7** | **12/7** | **Chalk and Black Bard** |
| **8.** | **Infix to postfix conversion using stack: algorithm and examples** | **17/7** | **15/7** | **Chalk and Black Bard** |
| **9** | **Evaluation of postfix expression: examples and algorithm****Multistack in one array: example and algorithm** | **18/7** | **17/7** | **Chalk and Black Bard** |
| **10** | **Queue: Introduction to queue, Queue as ADT, operations on queue.(queue menu driven program), Implementation of circular queue** | **19/7** | **18/7** | **Chalk and Black Bard** |
| **11** | **Implementation of priority queue (circular queue)** | **22/7** | **19/7** | **Chalk and Black Bard** |
| **12** | **Implementation of De-queue menu driven program.** | **24/7** | **22/7** | **Chalk and Black Bard** |
| **13** | **Applications of Queue, Introduction to Linked List, basic concept of linked list, memory allocation and de allocation of Linked List,**  | **25/7** | **24/7** | **Chalk and Black Bard** |
| **14** | **Program to implement of different functions on linked list like, insert(all three cases), display** | **26/7** | **25/7** | **Chalk and Black Bard** |
| **15** | **Implementation of different functions on linked list like, delete (all three cases), copy.** | **29/7** | **29/7** | **Chalk and Black Bard** |
| **16** | **implementation of different functions on linked list like, concatenate, split, reverse etc.** | **31/7** | **30/7** | **Chalk and Black Bard** |
| **17** | **Types of linked list, functions on doubly linked list like insert, delete, traverse forward and backward.** | **1/8** | **31/7** | **Chalk and Black Bard** |
| **18** | **Circular linked list and functions on it. Linked representation of stack and queue. Applications of linked list.** | **2/8** | **01/8** | **Chalk and Black Bard** |
| **19** | **Introduction to Trees, Definitions of different tree terminologies like leaf node, depth etc.**  | **05/8** | **01/8** | **Chalk and Black Bard** |
| **20** | **Binary tree representation, operations on binary tree, like create, count leaf nodes, count interior nodes etc. Tree traversal techniques like inorder, preorder, postorder (recursive)etc.** | **07/8** | **07/8** | **Chalk and Black Bard** |
| **21** | **Inorder and preorder: non recursive algorithm.****Examples and implementation.** | **08/8** | **08/8** | **Chalk and Black Bard** |
| **22** | **Postorder: non recursive algorithms, examples and implementation.** | **09/8** | **9/8** | **Chalk and Black Bard** |
| **23** | **Binary search Tree, examples, implementation of different operations on BST like insert, search, count leaf nodes, count interior nodes etc.** | **19/8** | **19/8** | **Chalk and Black Bard** |
| **24** | **Delete in BST: examples and implementation. Threaded Binary Tree. Applications of Trees.** | **21/8** | **21/8** | **Chalk and Black Bard** |
| **25** | **Expression Tress, Algorithm to construct binary tree from postfix expression. Examples on it. Applications of Trees.** | **22/8** | **22/8** | **Chalk and Black Bard** |
| **26** | **Introduction to Graph, definitions of Graph terminologies, Types of graphs.**  | **23/8** | **23/8** | **Chalk and Black Bard** |
| **27** | **Graph representation, Graph Traversal: Depth First Search(DFS): algorithm and examples**  | **26/8** | **26/8** | **Chalk and Black Bard** |
| **28** | **Breadth First Search (BFS): algorithm and examples. Minimum Spanning Tree: Definition and examples.** | **28/8** | **27/8** | **Chalk and Black Bard** |
| **29** | **Prim’s algorithm and examples.****Kruskal’s algorithm and examples.** | **29/8** | **28/8** | **Chalk and Black Bard** |
| **30** | **Dijkstra’s Shortest Path Algorithm and examples. Applications of graph.** | **30/9** | **9/9** | **Chalk and Black Bard** |
| **31** | **Introduction to Analysis, Algorithms, characteristics of an algorithms.** | **9/9** | **11/9** | **Chalk and Black Bard** |
| **32** | **Time and space complexities, Asymptotic notations and examples.** | **11/9** | **13/9** | **Chalk and Black Bard** |
| **33** | **Growth functions, order of growth functions, Analysis of algorithms having linear, quadratic,.** | **13/9** | **16/9** | **Chalk and Black Bard** |
| **34** | **Analysis of algorithms having cubic, logarithmic, exponential complexity.** | **16/9** | **18/9** | **Chalk and Black Bard** |
| **35** | **Introduction to sorting, types of sorting algorithms, Bubble sort, modified bubble sort.** | **18/9** | **19/9** | **Chalk and Black Bard** |
| **36** | **Selection sort, insertion sort, tracing of algorithms and examples. Analysis of bubble sort, selection sort and insertion sort.** | **19/9** | **20/9** | **Chalk and Black Bard** |
| **37** | **Quick sort algorithm, tracing and examples. Analysis of quick sort. (Best, Average and worst case analysis and examples.)** | **20/9** | **23/9** | **Chalk and Black Bard** |
| **38** | **Merge sort algorithm, tracing and examples. Analysis of merge sort.** | **23/9** | **25/9** | **Chalk and Black Bard** |
| **39** | **Heap sort algorithm, tracing and examples. Analysis of heap sort.** | **25/9** | **26/9** | **Chalk and Black Bard** |
| **40** | **Shell sort algorithm, tracing and examples. Analysis of shell sort. Radix sort algorithm.** | **26/9** | **30/9** | **Chalk and Black Bard** |
| **41** | **Tracing and examples of radix sort. Analysis of radix sort. Comparison of sorting algorithms.** | **27/9** | **3/10** | **Chalk and Black Bard** |
| **42** | **Introduction to Searching, Linear and Binary search examples.** | **30/9** | **04/10** | **Chalk and Black Bard** |
| **43** | **Hashing techniques, Different hash functions.****Collision and collision resolution techniques. Analysis of searching techniques.** | **3/10** | **07/10** | **Chalk and Black Bard** |
| **44** | **Revision and doubt solving** | **4/10** | **09/10** | **Chalk and Black Bard** |

Lab Plan: Data Structures Lab (ITL302)

Lab Outcomes: Students 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.

Lab Plan: Data Structures Lab

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Topic | Week No. | Lab Outcome |
| 1 | Implementation of stack using array | Week 1 | CO1, CO2, CO3, CO6 |
| 2 | Implementations of Infix to Postfix transformation and its evaluation. | Week 1 | CO1, CO2, CO3, CO6 |
| 3 | Implementations of double ended queue menu driven program. | Week 2 | CO1, CO2, CO3, CO6 |
| 4 | Implementation of different operations on singly linked list.  | Week 3 | CO1, CO2, CO3, CO6 |
| 5 | Implementation of different operations on doubly linked list.  | Week 4 | CO1, CO2, CO3, CO6 |
| 6 | Implementation of different Operations on BST  | Week 5 | CO1, CO2, CO3, CO6 |
| 7 | Implementation of construction of expression tree using postfix expression. | Week 6 | CO1, CO2, CO3, CO6 |
| 8 | Implementation of quick sort | Week 7 | CO4, CO6 |
| 9 | Implementation of merge sort | Week 7 | CO4, CO6 |
| 10 | Implementation of Heap Sort | Week 8 | CO4, CO6 |
| 11 | Implementation of hashing functions with different collision resolution techniques | Week 9 | CO4, CO5,CO6 |
| 12 | Implementation of priority queue | Week 10 | CO1, CO2, CO3, CO6 |
| 13 | Implementation of graph and its traversals, DFS and BFS | Week 11 | CO1, CO2, CO3, CO6 |

**Assignment Plan:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Assig****No.** | **Date** | **Questions** | **CO/LO** |
| **1** | **28/08/2019** | 1. Select and write appropriate data structure as applicable to following specified problem definitions.
2. Convert ((5+4)\*8/(3\*4)) into postfix using stack. Show status of stack at every step. (Draw pictures of different stack to show the status of stack).
3. Use the postfix expression in the question no. 2 and evaluate the same using stack. Show status of stack diagrammatically at every step.
4. Write a program to convert infix expression into prefix.
5. Write a program to implement operations of queue like insert and remove using stack as container.

**Hint**: insert will push element in stack container, but whenever remove operation is performed it should remove the first pushed element from stack container which is present at bottom in the stack container. **How many stacks are needed to implement queue using stack?**1. Write a C program to merge two sorted linked list into new linked list.
2. Write a C program to return the Nth node from the end of a linked list
3. Write functions to Print alternate nodes and remove duplicates in the linked list.
 | **CO1, CO2, CO3** |
| **2** | **10/10/2019** | 1. Trace the quick sort algorithm to sort the array

[11, 25,9,3,5,0,20,28,7]into**ascending** order. Use the array implementation **exactly** as described in the class.Also list the calls to **quicksort** and **partition** in the order they occur. Assume that the last element is chosen as pivot.2) Write a program in C to implement Radix sort and determine its complexity.Write a program in C to Implement binary search algorithm and determine its complexity.**3)** Compute the complexity merge sort algorithm with recurrence relation.**4)** Determine the running time analysis on quick sort, merge sort and insertion sort**5)** Here is an array of ten integers:5, 3, 8, 9, 7, 2, 6, 4Draw the above array after all the iterations of Heap sort. Show all intermediate steps.**6)** Compare all sorting algorithms on their running time complexities for best, average and worst case in tabular form.**7)** Define asymptotic notations along with examples and List out the properties of asymptotic notations.**8)** Draw a hash table with open addressing and a size of 9. Use the hash function "k%9". Insert the keys: 5, 29, 20, 0, 27 and 18 into your table (in that order). | **CO4, CO6** |

**Term Work:**

Term Work shall consist of at least 10 to 12 practical’s based on the above list. Also Term work

Journal must include at least 2 assignments.

**Term Work Marks:** 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5

Marks (Attendance)

**Oral & Practical Exam:** An Oral & Practical exam will be held based on the above syllabus.