## Course Plan

**Semester: III**

| | |
|---|---|
| *Course Title:D**ata Structures*** | *Course Code: 19IS305* |
| *L-T-P-SS (Hours/Week): 4-0-0-0* | *No. of Credits: 04* |
| *Total Lecture Hours: 52* | *Duration of SEE: 03 Hours* |
| *CIE Marks: 50* | *SEE Marks: 50* |

### *Prerequisites*

Students are expected to have to have a preliminary knowledge of C programming.

### *Course Overview*

A good algorithm usually comes together with a set of good data structures that allow the algorithm to manipulate the data efficiently. In this course, we consider the common data structures that are used in various computational problems. You will learn how these data structures are implemented using programming languages and will practice implementing them in our programming lab. This will help you to understand what is going on inside a particular built-in implementation of a data structure and what to expect from it.

### *Scope of the Course:*

- Basics of data structures including their fundamental building blocks: arrays and linked lists
- How to use Dynamic arrays .
- How to use priority queues to efficiently schedule jobs, in the context of a computer operating system or real life
- Basic structure of binary search trees - AVL trees and Splay trees
- Applications of data structures

## Course Learning Objectives (CLO)

After studying this subject, the student should be able to

- To understand and analyze various data organization.
- To identify and implement the appropriate data structure and modify it if required for modeling a given problem and perform various operations on it.
- To classify and examine linear and non linear data structures
- To demonstrate and practice iterative and recursive solutions for elementary problems.
- To formulate algorithms and programs that use data structures such as arrays, linked lists, stacks, queues, trees,graphs.

## Course Content

Subject Code**: 19IS305** L-T-P-SS**: 0-4-0-0**
Hours/Week : **4** CIE**: 50**
Teaching Hours**: 52 Hrs** SEE**: 50**
Exam Hours: **3** Credits : **4**

| | |
|---|---|
| **UNIT – I** | |
| **Chapter 1: Introduction And Overview**<br>Definitions, Concepts of data structures, types, Overview of data structures. | 1 Hrs |
| **Chapter 2: Pointers**<br>Definition and Concepts, Accessing variables through pointers, Pointers and functions, Arrays and pointers, Array of pointers, Pointer arithmetic. | 4 Hrs |
| **Chapter 3: Linear Data Structures - Stacks**<br>Introduction and Definition, Representation of stack: Array and structure representation, Operations on stacks, Applications of stack: Conversion of Expressions, Evaluation of expressions, Recursion: Implementation, Simulating Recursion, examples on Recursion. | 5 Hrs |
| **UNIT – II** | |
| **Chapter 4: Linear Data Structures - Queues**<br>Introduction and Definition Representation of Queue: Array and Structure representation of queue, various queue Structures: ordinary queue, circular queue, priority queue. | 4 Hrs |
| **Chapter 5: Linear Data Structures - Singly Linked lists**<br>Memory allocation functions. Definition and concepts singly Linked List: Representation of link list in memory, Operations on singly Linked List , Circular Linked List. | 6 Hrs |
| **UNIT – III** | |
| **Chapter 6: Linear Data Structures - Doubly Linked lists**<br>Doubly Linked List: Representation and Operations, Circular doubly Link list: Representation and Operations. Linked List representation of stack, Linked List representation of queue. | 6 Hrs |
| **Chapter 7: Nonlinear Data Structures: Tree data structures 1**<br>Basic Terminologies, Binary Trees: Properties, Representation of Binary Tree: Linear representation, Linked representation, Operations on Binary Tree: Insertion, Simple Deletion, Traversals. Binary Search Tree, Operations on Binary Search Tree: Insertion, Traversals. | 6 Hrs |
| **UNIT – IV** | |
| **Chapter 8: Nonlinear Data Structures - Tree data structures 2**<br>Expression Tree: Evaluating expression tree, Constructing expression tree from | 10Hrs |

| | |
|---|---|
| postfix expression, traversals, Threaded binary Tree: types, B-Trees, B+ Trees, AVL Trees: Definition, Constructing a general AVL tree. | |
| **UNIT – V** | |
| **Chapter 9: Nonlinear Data Structures - Graph data structures**<br>Graph terminologies: Walks, Paths, Circuits, Connected graphs, Disconnected graphs and Components, Euler graphs. Directed graphs, Undirected graphs, Hamiltonian paths and Circuits | 5 Hrs |
| **Chapter 10: Representation of graphs**<br>Set Representation, Linked representation, Matrix representation. Operations on Graphs: Insertion and Deletion of edges and vertices (linked representation), DFS, BFS, | 5 Hrs |

**Text Books:**

1. **Fundamentals of Data Structures In C;** 2nd Edition; Ellis Horowitz, Anderson-Freed, Sahni ; Universities Press, 2008.
2. **Data Structure using C**; Aaron M. Tenenbaum, Yedidyah Langsam & Moshe J. Augenstein; Pearson Education/PHI, 2006.
3. **Data Structures With C**; Seymour Lipschutz ; Tata Mcgraw Hill Education Private Limited 2010.
4. **Data Structures Using C And C++**; 2nd Edition; Langsam Yedidyah, Augenstein Moshe J., Tenenbaum Aaron M; Prentice-Hall 2009.

**Reference Books:**

1. **Classic Data Structures;** D.Samanta.
2. **Data Structures and Program Design in C**; R. Kruse etal, , Pearson Education Asia, Delhi-2002.
3. **Computer Science A Structured Programming Approach Using C;** Second Edition; Behrouz A. Forouzan and Richard F. Gilberg;Thomson, 2003.

**E-RESOURCES:**

1. https://onlinecourses.nptel.ac.in/noc16_cs06/
2. www.tutorialspoint.com/cprogramming/c_pointer

**Evaluation Scheme**

**CIE Scheme :**

| *Assessment* | *Marks Distribution* |
|---|---|
| Mid Semester Exam 1 | 20 |
| Mid Semester Exam 2 | 20 |
| Task 1: Design Assignment | 03 |
| Task 2: Quiz or surprise test | 03 |
| Task 3:Project | 04 |
| Total | 50 |

Semester End Examination (SEE) is a written examination of three hours duration of 100 marks.

**Subject Utilization for Mid Semester Exams (MSE) and Semester End Examination(SEE)**

| Unit | Chapter | | Teaching Hours | No of Questions in | | |
|---|---|---|---|---|---|---|
| | | | | MSE 1 | MSE 2 | SEE |
| 1 | 1 | Introduction and Overview | 01 | 01 | --- | 01 |
| | 2 | Pointers | 04 | | | |
| | 3 | Stacks | 05 | 01 | --- | 01 |
| 2 | 4 | Queues | 04 | 01 | --- | 01 |
| | 5 | Singly Linked Lists | 06 | 01 | --- | 01 |
| 3 | 6 | Doubly Linked Lists | 06 | --- | 01 | 01 |
| | 7 | Trees 1 | 06 | | 01 | 01 |
| 4 | 8 | Trees 2 | 10 | --- | 02 | 02 |
| 5 | 9 | Graphs | 05 | --- | --- | 01 |
| | | Graph representations | 05 | | | 01 |

Note

- Each question carries 20 marks and may consist of sub-questions.
- Mixing of sub-questions from different chapters within a unit (only for Unit I, Unit II, Unit III and Unit IV) is allowed in Mid Semester Exam I, II and SEE.
- Answer any 5 full questions of 20 marks each by selecting one question from each unit in SEE.

## Course outcomes

After going through this course the students will be able to:

| Sl. No | Course Outcomes | Bloom's Taxonomy Level (BTL) |
|---|---|---|
| C305.1 | Understand the concepts of data structure, data type and array data structure, pointers, stacks. | L3 |
| C305.2 | Apply the concept of queues and singly linked list data structure to solve problems. | L3 |
| C305.3 | Apply the concept of circular and doubly linked list data structure | L3 |
| C305.4 | Apply the concept of Trees to solve problems | L3 |

| | | |
|---|---|---|
| C305.5 | Apply the concept of threaded binary tress,study hashing techniques and Graphs. | L3 |

**Mapping of POs & COs:**

| Table: Mapping of COs to PIs, POs and BTL | | | |
|---|---|---|---|
| **Course Outcomes (COs)** | **Program Outcomes (POs) Addressed** | **Performance Indicators (PI)** | **Bloom's Taxonomy Level (BTL)** |
| CO1 | 1,2,3 | 1.3,1.4,2.1.2,2.4.3,3.2.1 | L3 |
| CO2 | 1,2,3 | 1.3,1.4,2.1.2,2.4.3,3.2.1 | L3 |
| CO3 | 1,2,3 | 1.3,1.4,2.1.2,2.4.3,3.2.1 | L3 |
| CO4 | 1,2,3 | 1.3,1.4,2.1.2,2.4.3,3.2.1 | L3 |
| CO5 | 1,2,3 | 1.3,1.4,2.1.2,2.4.3,3.2.1 | L3 |

**Mapping of PO s & Cos:**

| **POs COs** | **PO 1** | **PO 2** | **PO 3** | **PO 4** | **PO 5** | **PO 6** | **PO 7** | **PO 8** | **PO 9** | **PO 10** | **PO 11** | **PO 12** | **PSO 1** | **PSO 2** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C305.1 | 1 | 2 | | | | | | | | | | | 2 | 2 |
| C305.2 | 1 | 2 | | | | | | | | | | | 2 | 2 |
| C305.3 | 1 | 2 | | | | | | | | | | | 2 | 2 |
| C305.4 | 1 | 2 | | | | | | | | | | | 2 | 2 |
| C305.5 | 1 | 2 | | | | | | | | | | | 2 | 2 |

**(1 = Low 30%-49%, 2 = Medium 50%-69%, 3 = High >70%)**

**Evaluation Scheme:**

**Bloom's Taxonomy Levels (BTL) Planned in MSE and SEE:**

| Sl. No. | Bloom's Taxonomy Levels (BTL) | MSE 1 | MSE 2 | MSE 3 | SEE |
|---|---|---|---|---|---|
| 1 | Remember | 10% | 10% | 10% | 10% |
| 2 | Understand | 15% | 15% | 15% | 15% |
| 3 | Apply | 30% | 30% | 30% | 30% |
| 4 | Analyze | 25% | 25% | 25% | 25% |
| 5 | Evaluate | 20% | 20% | 20% | 20% |
| 6 | Create | 00% | 00% | 00% | 00% |

## UNIT – I

## Chapter Wise Plan

| **Unit 1** | **10 Hours** |
|---|---|
| **Chapter 1: Introduction And Overview** | **01 Hours** |
| 1. Definition Types & Needs of Data Structures, Linear & Non linear Data Structures. | 01 Hour |

**Topic Learning Objectives (TLO):**

| Sl. No. | Topic Learning Objective | CO Mapped | BTL |
|---|---|---|---|
| 1 | Use the concept of structures in programming. | CO 1 | 3 |
| 2 | Explain the need for structures. | CO 1 | 3 |
| 3 | Understand the types of data structure | CO 1 | 3 |

**Review Questions:**

| Sl. No. | Question | Marks | CO Mapped | BTL |
|---|---|---|---|---|
| 1 | What is data structure? | 04 | CO 1 | 3 |
| 2 | List out the areas in which data structures are applied extensively? | 06 | CO 1 | 3 |
| 3 | Classify linear data structure and non linear data structure? | 06 | CO 1 | 3 |
| 4 | Give two examples of linear data structures. | 04 | CO3 | 3 |
| 5 | What are three examples of nonlinear data structures? | 06 | CO3 | 3 |
| 6 | What is the difference between the logical representation of a data structure and the physical representation? | 05 | CO2 | 3 |

**Lesson Schedule:**

| **Unit 1** | **10 Hours** |
|---|---|
| **Chapter 2: Pointers** | **02 Hours** |
| 1. Definition and pointer Concepts | 01 Hour |
| 2. Accessing variables through pointers<br>3. Arrays of pointers<br>4. Pointer Arithmetic | 01 Hour |

**Topic Learning Objectives (TLO):**

| Sl. No. | Topic Learning Objective | CO Mapped | BTL |
|---|---|---|---|
| 1 | Understand the concept of pointer arrays. | CO 1 | 3 |
| 2 | Perform pointer arithmetic operations . | CO 1 | 3 |
| 3 | Explain how to access structure members using . and ->operators. | CO 1 | 3 |

**Review Questions:**

| Sl. No. | Question | Marks | CO Mapped | BTL |
|---|---|---|---|---|
| 1 | Write a program to find the sum of elements of array elements using pointers. | 06 | CO 1 | 3 |
| 2 | Can we have Array of Pointers? Justify your answer with an example. | 06 | CO 1 | 3 |
| 3 | What is the output of the following program:<br>Main()<br>{<br>Int i=3,*j,**k;<br>j=&I;<br>k=&j;<br>printf(" address of i=%u,&i);<br>printf(" address of i=%u,j);<br>printf(" address of i=%u,*k);<br>printf(" address of j=%u,&j);<br>printf(" address of j=%u,k);<br>printf(" value of j=%u,j);<br>printf(" value of i=%d,*(&i));<br>} | 08 | CO 1 | 3 |

**Lesson Schedule:**

| **Unit 1** | **10 Hours** |
|---|---|
| **Chapter 3: Stack & Recursion** | **07 Hours** |
| 1. Stack definitions & concepts, Representing Stacks<br>2. Operations on stacks, Applications of stack<br>3. Recursion: Implementation, Simulating Recursion<br>4. Examples on Recursion | 05 Hours<br><br>02 Hours |

**Topic Learning Objectives (TLO):**

| Sl. No. | Topic Learning Objective | CO Mapped | BTL |
|---|---|---|---|
| 1 | Understand & Implement Stack | CO 1 | 3 |
| 2 | Analyze different representations for stack | CO 1 | 2 |
| 3 | Evaluate expressions using stack | CO 1 | 2 |
| 4 | Converting infix expression to postfix | CO 1 | 3 |

**Review Questions:**

| Sl. No. | Question | Marks | CO Mapped | BTL |
|---|---|---|---|---|
| 1 | Explain the procedure to convert a valid infix expression to postfix expression with algorithm. Trace the algorithm for the following infix expression<br><br>(A + B) + ((C - D) * E + (F - G)) | 4 | CO 1 | 3 |
| 2 | Write a C program for the following operations on a stack of integers (use array).<br>a) Push<br>b) Pop<br>c) Display | 6 | CO 1 | 3 |
| 3 | Write a C program to covert and print a given valid parenthesized infix arithmetic expression to postfix expression. | 4 | CO 1 | 3 |
| 4 | Write a program to implement Stack using Array. | 6 | CO 1 | 3 |

**Lesson Schedule:**

| **Unit 2** | **10 Hours** |
|---|---|
| **Chapter 4: Queue** | **05 Hours** |
| 1. Introduction and Definition<br>2. Representation of Queue: Array and Structure representation of<br>3. Various queue Structures: ordinary queue, circular queue, priority queue. | 03 Hours<br><br>02 Hours |

**Topic Learning Objectives (TLO):**

| **Sl. No.** | **Topic Learning Objective** | **CO Mapped** | **BTL** |
|---|---|---|---|
| 1 | Understand and **Define** queue & its types. | CO 2 | 2 |
| 2 | **Differentiate & Implement** basic queue operations(insert, delete ) with respect to different types of queues. | CO 2 | 3 |
| 3 | **Applications** of queue. | CO 2 | 3 |

**Review Questions:**

| **Sl. No.** | **Question** | **Marks** | **CO Mapped** | **BTL** |
|---|---|---|---|---|
| 1 | Choose an appropriate representation for linear queue in C and implement the following routines in C for a queue of integers:<br>i) Insert ii) Delete | 04 | CO 2 | 3 |
| 2 | Implement the routines insert ()and delete() in C for circular queue. | 06 | CO 2 | 3 |
| 3 | Show how to sort a set of numbers using a priority queue and operations insert and delete. | 06 | CO 2 | 3 |

**Lesson Schedule:**

| **Unit 2** | **10 Hours** |
|---|---|
| **Chapter 5: Linear Data Structures - Singly Linked lists** | **05 Hours** |
| 1. Memory allocation functions.<br>2. Definition and concepts singly Linked List<br>3. Representation & Operations on singly Linked List, Circular linked list. | 02 Hours<br><br>03 Hours |

**Topic Learning Objectives (TLO):**

| Sl. No. | Topic Learning Objective | CO Mapped | BTL |
|---|---|---|---|
| 1 | Understand advantages and disadvantages of Array over Linked list | CO 2 | 2 |
| 2 | Understand different Memory Allocation Functions | CO 2 | 2 |
| 3 | Define and Represent the principles of Singly Linked List. | CO 2 | 3 |
| 4 | Implement various operations on Singly Linked List. | CO 2 | 3 |

**Review Questions:**

| Sl. No. | Question | Marks | CO Mapped | BTL |
|---|---|---|---|---|
| 1 | Implement the routines insert and delete for a singly linked list | 06 | CO 2 | 3 |
| 2 | Implement queue using linked list | 06 | CO 2 | 3 |
| 3 | What are the advantages and disadvantages of representing a group of items as an array versus a linear linked list? | 06 | CO 2 | 3 |
| 4 | Design a program that reads the name, age and salary of 10 persons and maintain them in a linked list sorted by name. | 08 | CO 2 | 3 |
| 5 | What is dynamic memory allocation? Explain with an example | 06 | CO 2 | 3 |
| 6 | Explain the difference between malloc and calloc functions used to allocate memory dynamically with suitable examples. | 08 | CO 2 | 3 |

***Model Question Paper for MSE-I***

**NMAM INSTITUTE OF TECHNOLOGY, NITTE**
**(An Autonomous Institution affiliated to VTU, Belgaum)**
**III SEM BE (ISE) Mid Semester Examinations-I**
**17IS305 - Data Structures**

**Time: 1 Hour** **Marks: 20**

**Note: Answer any one full question from each Unit**

**Unit-I**

1. a) What is the output of the following program? **(2)**

```
#include <stdio.h>
Void main()
{int a[5]={10,20,30,40,50,}
 Printf (%p%p\n",&a[0],a,a+0);
```

}

b) Write a program using pointers to structure to display the details of n books with fields (author, name, price, edition). **(8)**

2. a) What is the need for converting and infix expression to postfix or prefix expression? (2)
b) Write an algorithm to convert an infix expression to postfix expression.
Trace that algorithm and obtain the postfix expression for the following:
((A+ (B*C)) ^E-F/D). **(8)**

**Unit-II**

3. a) Consider that a singly linked list contains the following elements: (6)
*book number: integer, Author Name: string of max 25 char, cost: float*
Write a program in C to represent a singly linked list with the above elements, using separate functions insert an element at the front, delete an element at the end, display the list.

b) Perform the following operations in a circular linked queue of size 5. Insert the elements 10, 20, 30, delete, delete, insert 40, 50, 60, 70, 80 . Properly show the positions of front and rear pointers. (4)

4. a) What do you understand by term dynamic memory allocation? Explain any two functions that support dynamic allocation. **(4)**
b) Write a C program to implement circular queue using functions to **(6)**
i) insert at rear ii) delete at front iii) Display
(Pass parameters to functions).

**Lesson Schedule:**

| **Unit 3** | **12 Hours** |
|---|---|
| **Chapter 6:Linear Data Structures – Doubly Linked lists & Trees** | **12 Hours** |
| 1. Doubly Linked List: Representation and Operations<br>2. Linked List representation of stack, Linked List representation of queue.<br>3. Basic terminologies of trees and traversals | 03 Hours<br>04 Hours<br>05 Hours |

**Topic Learning Objectives (TLO):**

| Sl. No. | Topic Learning Objective | CO Mapped | BTL |
|---|---|---|---|
| 1 | Define and understand the principles of Doubly Linked List | CO 3 | 2 |
| 2 | Create and Perform various operations on Doubly Linked List | CO 3 | 3 |
| 3 | Implement Stack and Queue using Linked List | CO 3 | 3 |
| 4 | Create binary search tree | CO 3 | 3 |

**Review Questions:**

| Sl. No. | Question | Marks | CO Mapped | BTL |
|---|---|---|---|---|
| 1 | Write a C/C++ program to search a node with value x and delete the node if it is found from a doubly linked list. | 06 | CO 3 | 3 |
| 2 | List out the advantage and disadvantage of doubly linked list over singly linked list | 06 | CO 3 | 3 |
| 3 | How tree traversal is done? | 08 | CO 3 | 3 |

**Lesson Schedule:**

| **Unit 4** | **10 Hours** |
|---|---|
| **Chapter 7:**<br>**Nonlinear Data Structures: Tree data structures 2** | **10 Hours** |
| 1. Expression tree<br>2. Tree traversals<br>3. B – Trees<br>4. B+ trees<br>5. AVL Trees | 02 Hours<br>02 Hours<br>02 Hours<br>02 Hours<br>02 Hours |

**Topic Learning Objectives (TLO):**

| Sl. No. | Topic Learning Objective | CO Mapped | BTL |
|---|---|---|---|
| 1 | Understanding expression tree concept | CO 4 | 2 |
| 2 | Understand and Implement basic B Trees | CO 4 | 3 |
| 3 | Create AVL Trees | CO 4 | 3 |

**Review Questions:**

| Sl. No. | Question | Marks | CO Mapped | BTL |
|---|---|---|---|---|
| 1 | Construct a B tree and B+ tree from: 20,34,67,11,55,69,204,190,123,8 | 06 | CO 4 | 3 |
| 2 | Write a function to construct expression tree using postfix expressions | 06 | CO 4 | 3 |
| 3 | CConstruct an AVL tree, from: 20,34,67,11,55,69,204,190,123,8 | 06 | CO 4 | 3 |
| 4 | Write a short note on threaded binary trees. | 08 | CO 4 | 3 |

*Model Question Paper for MSE-II*

**NMAM INSTITUTE OF TECHNOLOGY, NITTE**
**(An Autonomous Institution affiliated to VTU, Belgaum)**
**III SEM BE (ISE) Mid Semester Examinations-2**
**17IS305-Data Structures**

**Time: 1 Hour** **Marks: 20**

**Note: Answer any One full question from each Unit**

**Unit-I**

1. a) Write a function to count the number of nodes present in a doubly linked list. **(3)**
   b) Write a function to search an element in a DLL with appropriate validation messages. **(4)**
   c) Write a function to delete a node from the rear end in a DLL with an example. **(3)**

2. a) Write an algorithm to create a binary tree and perform pre order and post order traversals on it **(5)**
   b) Write a function to create BST. **(5)**

**Unit-II**

3. a)Write a function to construct expression tree using postfix expressions **(4)**
   b)construct an AVL tree, B tree and B+ tree from: 20,34,67,11,55,69,204,190,123,8. **(6)**
4. a) List the advantages and disadvantages of doubly linked lists over singly linked lists. **(4)**
   b)Write a short note on threaded binary trees. **(6)**

**Lesson Schedule:**

| **Unit 5** | **10 Hours** |
|---|---|
| **Chapter 8:Non linear Data Structures Graph** | **10 Hours** |
| 1. Graph Terminologies<br>2. Euler Graph<br>3. Circuits | 05 Hours |
| 4. Insertion and deletion of edges<br>5. DFS<br>6. BFS | 05 Hours |

**Topic Learning Objectives (TLO):**

| Sl. No. | Topic Learning Objective | CO Mapped | BTL |
|---|---|---|---|
| 1 | Understand basic terminologies in Graphs: Walks, Paths, and Circuits. | CO 5 | 1 |
| 2 | Distinguish Connected and Disconnected Graphs. | CO 5 | 2 |
| 3 | Understand various Graph Components. | CO 5 | 1 |
| 4 | Describe Hamiltonian Paths and Circuits. | CO 5 | 3 |

**Review Questions:**

| Sl. No. | Question | Marks | CO Mapped | BTL |
|---|---|---|---|---|
| 1 | Define following terms with proper examples: path, circuit, walk and loop. | 06 | CO 5 | 1 |
| 2 | Draw an undirected graph with five edges and four vertices'. The vertices should be called v1, v2, v3 and v4and there must be a path of length three from v1 to v4. Find out if there is a Hamiltonian circuit in the graph? | 06 | CO 5 | 3 |
| 3 | With an example explain what is connected and disconnected graph. | 08 | CO 5 | 2 |

***Model Question Paper for SEE***

**NMAM INSTITUTE OF TECHNOLOGY, NITTE**
**(An Autonomous Institution affiliated to VTU, Belgaum)**
**III SEM BE (ISE) Semester End Examinations Section: A,B&C**
**17IS305- Data Structures**

**Time: 3 Hrs** **Marks: 100**

**Note: Answer Five full questions choosing one full question from each unit.**

**Unit-I**

1.
   a. Write a recursive function to find the GCD of 2 numbers. 4
   b. Write a program to sort the elements in an array using ponters. 4
   c. Show how to use
      i. Pointer to structure
      ii. Array of pointers
      iii. Dynamic memory allocation 12

2.

a. Write an algorithm to implement a stack of size 20 using an array. The elements in the stack are integers. The operations to be supported are PUSH, POP and DISPLAY. Take into account the exception of stack overflow and stack underflow. 10

b. Write a C function to convert a valid arithmetic infix expression to its equivalent postfix expression and hence convert the following infix expression to postfix: ((A+B)*C-(D/E))$(F$G). 10

**Unit-II**

3.

a. Write the algorithms for following in a circular queue.
   i. cqinsert()
   ii. cqdelete()
   iii. cqdisplay() 10

b. Discuss the drawback of ordinary queue. How can you overcome it? Write a C++ program to create and find the sum of all numbers in an ordinary queue. 10

4.

a. What is a Linked list? Write a C++ program to insert a new node considering all possible cases. 10

b. Write a C++ program to delete a node at the front and back considering all possible cases. 10

**Unit-III**

5.

a. What are the advantages and disadvantages of doubly linked list? Write a C function to add a new node in doubly linked list considering all possible cases. 10

b. What are the advantages of circular linked list over linear list? Write a C routine to concatenate two circular lists. 10

6.

a. Write functions to create a binary tree given
   i. The preorder and inorder traversals of that tree.
   ii. The preorder and postorder traversals of that tree. 8

b. Write routines to traverse a binary tree in preorder and postorder? 4

c. Write a program to insert new nodes to a binary search tree and delete a node from binary search tree. 8

**Unit-IV**

7.

a. Write a short note on the following
   i. AVL tree
   ii. B-trees 8

b. Write a function to create a left-threaded binary tree. 6

c. Write algorithm to evaluate expression tree? 6

8. a. Construct B+ tree from the following: 10, 20, 30, 40, 50, 60, 70, 80, 90. 10

b. Write a function to create in-threaded binary tree. 6

c. Construct an AVL tree out of following: 34, 66, 78, 90, 12, 34, 54, 65, 76. 4

**Unit-V**

9.

a. Write a note on the following
   i. Walks
   ii. Paths
   iii. Circuits
   iv. Connected graphs
   v. Disconnected graphs 10

b. Explain the linked representation of a node? Give routine for adding a node to the linked representation of a graph? 10

10.

a. What is the difference between a tree and a graph data structure? Explain? 10

b. How do we find Hamiltonian circuit in graphs? Explain what is Euler graphs with example. 10