- **Sem.** : 3
- **Subject Code** : 05BC1303
- **Subject** : Data Structure
- **Course Objectives** :

1. To impart a thorough understanding of linear data structures such as stacks, queues and their applications.
2. To impart a thorough understanding of non - linear data structures such as trees, graphs and their applications.
3. To impart a thorough familiar with writing recursive methods.
4. To design and implement various data structure algorithms.
5. To introduce various techniques for representation of the data in the real world.

- **Prerequisites** : Functional Knowledge of c programming language

| Unit No | Topics Covered | No of lectures required |
|---|---|---|
| 1 | **Introduction of data structure :** Introduction of data and data type, introduction of data structure, primitive and non-primitive data structure, Define Complexity of Data structure – Time and Space complexity, best case, worst case and average case. | 6 |
| 2 | **Stacks and Queues :** Stack – introduction, operations, applications of stack recursion and polish notation Queue – introduction, types of queue, simple queue and its operations, circular queue and its operations, real life applications of queue | 10 |
| 3 | **Linked List :** Introduction, types of linked list – singly, doubly and circular, stack and queue using linked list. | 12 |
| 4 | **Tree :** Basic terminologies, M-ary tree, binary tree – array and linked representation, operations and traversal, conversion of general tree to binary tree, threaded binary tree, binary search tree | 12 |

| | | |
|---|---|---|
| **5** | **Graph :**<br>Basic terminologies, representations of graph – adjacency matrix and adjacency list, BFS and DFS traversal | **10** |

- **Course Outcomes:**

1. Compare different data structures. Pick an appropriate data structure for a design situation.
2. Use appropriate data structures like arrays, linked list, stacks and queues to solve real world problems efficiently.
3. Represent and manipulate data using nonlinear data structures like trees and graphs to design algorithms for various applications.
4. Implement operations like searching, insertion, and deletion, traversing mechanism etc. on various data structures.
5. Determine and analyze the complexity of given Algorithms.

- **Course Outcomes – Program Outcomes Mapping Table :**

| | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | H | H | L | - | M | - | - | L | H | - | H |
| **CO2** | H | H | L | - | M | - | - | - | H | - | M |
| **CO3** | H | H | L | - | M | - | - | - | H | - | M |
| **CO4** | H | H | L | - | L | - | - | - | H | - | M |
| **CO5** | H | H | L | - | H | - | - | L | H | - | H |

- **Text Book :**

1. Reema Thareja. “Programming in C”, Oxford University Press.

- **Reference Books :**

1. "Introduction to Algorithm", Cormen, Leiserson, Rivest, Stein, , PHI (2003), 2nd Edition,
2. "Design and Analysis of Algorithms" Parag Dave & Himanshu Dave, Pearson Education (2008).
3. “Data Structures using C”, A. K. Sharma, Pearson Education (2011).
4. "Data Structures: A Pseudo-code Approach with C", Gilberg & Forouzan, , Cengage Learning.
5. “Fundamentals of Data Structures in C”, Horowitz, Sahni, Anderson-Freed, University Press (2nd edition-2007)
6. "Data Structures Using C & C++", Tenenbaum, PHI. (Mention atleast 3 reference books)

- **Web References :**
  1. https://www.geeksforgeeks.org/data-structures/
  2. https://www.javatpoint.com/data-structure-tutorial
- **App References:**
  1. Data Structures and Algorithms offline Tutorial - ONAN Mobile Software
  2. Data Structure Using C - Super Dream
- **Syllabus Coverage from text /reference book & web/app reference:**

| Unit No | Chapter Numbers |
|---|---|
| 1 | Chapter-2 |
| 2 | Chapter-7 and Chapter-8 |
| 3 | Chapter-6 |
| 4 | Chapter-9 |
| 5 | Chapter-13 |

# PRACTICALS

| Unit No | List of Practicals |
|---|---|
| 1 | 1.1 Create an array of size 10, input values and print the array, and search an element in the array.<br>1.2 Create an array of size 10, input values and display sum and average of all elements in the array.<br>1.3 Create arrays A, B and C of size 3, perform C = A + B.<br>1.4 Create arrays A, B of size 3, C of size 6, merge A and B into C.<br>1.5 Create an array of size 10, find the largest value from the array.<br>1.6 Insert an element into the array at user defined position.<br>1.7 Delete an element from the array from user defined position.<br>1.8 Sort the array into ascending order.<br>1.9 Sort the array into descending order.<br>1.10 Write a program to multiply two matrices. |
| 2 | 2.1 Implement stack using array with following operations:<br>push, pop, print, peek, peep, change, exit.<br>2.2 Write a program to find out factorial of number using recursion (stack).<br>2.3 Write a program to print string in reverse order using stack.<br>2.4 Write a program to find factorial of a given integer number using stack.<br>2.5 Write a program to find the power of a given number using stack.<br>2.6 Write a program to find GCD of given two numbers.<br>2.7 Write a program to find Smallest Common Divisor of a given number.<br>2.8 Write a program find Minimum and Maximum number from the given array using Recursion.<br>2.9 Write a program which performs following operations using simple queue. : insert() -> delete() -> display()<br>2.10 Write a program which performs following operations using circular queue. :insert() -> delete() -> display() |
| 3 | 3.1 Write a program to perform following operation on singly linked list:<br>a. Create a linked list<br>b. Display it<br><br>3.2 Write a program to perform following operation on singly linked list:<br>a. insert a node at the starting of the list<br>b. insert a node at the end of the list |

| | |
|---|---|
| | 3.3 Write a program to perform following operation on singly linked list:<br>a. insert a node after the specific node<br>b. insert a node before the specific node<br><br>3.4 Write a program to perform following operation on Doubly linked list:<br>a. Create a linked list<br>b. Display it<br><br>3.5 Write a program to perform following operation on Doubly linked list:<br>a. insert a node at the starting of the list<br>b. insert a node at the end of the list<br><br>3.6 Write a program to perform following operation on Doubly linked list:<br>a. insert a node after the specific node<br>b. insert a node before the specific node<br><br>3.7 Write a program to perform following operation on Circular Singly linked list:<br>a. Create a linked list<br>b. Display it<br><br>3.8 Write a program to perform following operation on Circular Singly linked list:<br>a. insert a node at the starting of the list<br>b. insert a node at the end of the list<br><br>3.9 Write a program to perform following operation on Circular Singly linked list:<br>a. insert a node after the specific node<br>b. insert a node before the specific node<br><br>3.10 Write a program to count the number of nodes available in the Singly linked list created before. |
| **4** | 4.1 Write a program to create a binary tree . Traverse tree in preorder , postorder and inorder.<br><br>4.2 Write a program to perform following operations on Binary search tree:<br>a. insert<br>b. delete<br>c. height of the tree<br>d. depth of the tree<br>e. total no. of nodes in the tree |