C AND DATA STRUCTURES

COURSE CODE: 20CA3101 **L T P C**

3 1 0 4

COURSE OUTCOMES:

At the end of the course, student will be able to
CO1: Apply the concept of arrays for storing, sorting and searching data
CO2: Apply the concept of pointers for dynamic memory management and string handling
CO3: Develop programs for searching and sorting.
CO4: Develop programs using concepts of trees.
CO5: Apply concepts of graphs

UNIT-I **(10 Lectures)**
An Overview of C, Basic Data types, modifying the basic data types, identifier names, Variables, type Qualifiers, Constants, Operators, Expressions.
Selection, Iteration and Jump Statements
Functions: Designing Structured Programs, Functions Basics, Standard Library Functions, User Defined Functions, Categories of functions, Parameter passing techniques, Scope, Scope Rules, Storage Classes and Type Qualifiers.
Recursion: Recursive Functions, Preprocessor directives.
Arrays: Concepts, Using Arrays in C, Inter-Function Communication using arrays, Array Applications, Two Dimensional Arrays, Introduction to Multidimensional Arrays,
Strings – Concepts, C Strings, String Input/output Functions, Arrays of Strings, String Manipulation functions.

Learning Outcomes:
At the end of the module, the student will be able to
● Apply the concepts of arrays in solving problems (L1)
● Develop programs that mimics string functions in solving problems (L2)

UNIT-II **(10 Lectures)**
Pointers: Introduction, Pointer Arithmetic, Pointers for Inter-Function Communication, Pointers to Pointers,
Arrays and Pointers- Array of Pointers, Pointer to Array, Pointers to void, Pointers to Functions, Command Line Arguments.
Dynamic Memory Allocation Functions, Programming Applications.

Learning Outcomes:
At the end of the module, the student will be able to
● Apply the concepts of pointers with respect to arrays and functions (L2)
● Demonstrate the usage of dynamic memory allocation functions to solve problems (L2)

UNIT-III **(10 Lectures)**
Stacks: Introduction, stack operations, applications.
Queues: Introduction, Operations on queues, circular queues, Priority queues, applications.
Linked Lists:

Introduction, Singly linked lists, circular linked lists, doubly linked lists, multiple linked lists, applications.
Linked stacks and linked queues:
Introduction, operations on linked stacks and linked queues, dynamic memory management, implementation of linked representations, applications.

Learning Outcomes:
At the end of the module, students will be able to

- Describe the advantages of linked implementation over array implementation of various data structures.(L3)
- Demonstrate how to declare structures to be used in simple linked lists, double linked lists, and circular linked lists. (L3)
- Implement the algorithms for inserting, deleting, and searching in a simple linked list. (L3)

UNIT-IV **(10 Lectures)**
Searching:
Introduction, linear search, , binary search, Fibonacci search.
Internal sorting:
Introduction, bubble sort, insertion sort, Quick sort
Trees and binary trees:
Introduction, Trees: definition and basic terminologies, representation of trees. Binary trees: basic terminologies and types, representation of binary trees, binary tree traversals, applications.

Learning Outcomes:
At the end of the module, students will be able to

- Explain what is meant by a balanced binary tree and why it is important. (L4)
- Demonstrate how to declare structures to be used in binary trees. (L4)
- Implement the algorithms for inserting, deleting, and searching for nodes in a binary tree. (L4)
- Demonstrate Searching and sorting techniques. (L3)

UNIT-V **(10 Lectures)**
GRAPHS:
Introduction, definitions and basic terminologies, representations of graphs, Minimal Spanning Trees graph traversals (BFS, DFS) and applications.

Learning Outcomes:
At the end of the module, students will be able to

- Discuss a basic search algorithm for graphs. (L3)
- Define minimal spanning tree and discuss, with diagrams, Prim's algorithm for finding the minimalspanning tree of a graph. (L5)
- Define shortest path and discuss, with diagrams, Dijkstra's algorithm for finding the shortest path fromnode x to node y of a graph. (L5)

TEXTBOOKS:

1. G.A.V PAI, Data Structures and Algorithms, Concepts, Techniques and Applications,Volume 1,1st Edition,Tata McGraw-Hill, 2008.

2. Richard F. Gilberg&Behrouz A. Forouzan, Data Structures, Pseudo code Approach with C, 2nd Edition, Cengage Learning India Edition, 2007.

REFERENCES:

1. Langsam, M. J. Augenstein, A. M. Tanenbaum, Data structures using C and C++, 2nd Edition, PHIEducation, 2008.
2. SartajSahni, Ellis Horowitz, Fundamentals of data Structures in C, 2nd Edition, Orient Blackswan, 2010.

WEB REFERENCES:

1. http://nptel.iitm.ac.in/video.php?subjectId=106105085.
2. https://www.cs.usfca.edu/~galles/visualization/Algorithms.html.