

## DATA STRUCTURES USING C

Semester	Course Code	Course Title	Hours	Credits
<b>II</b>	<b>C2</b>	<b>DATA STRUCTURES USING C</b>	<b>60</b>	<b>3</b>

### Course Objectives

To introduce the fundamental concept of data structures and to emphasize the importance of various data structures in developing and implementing efficient algorithms.

### Course Learning Outcomes:

Upon successful completion of the course, a student will be able to:

1. Understand available Data Structures for data storage and processing.
2. Comprehend Data Structure and their real-time applications - Stack, Queue, Linked List, Trees and Graph
3. Choose a suitable Data Structures for an application
4. Develop ability to implement different Sorting and Search methods
5. Have knowledge on Data Structures basic operations like insert, delete, search, update and traversal
6. Design and develop programs using various data structures
7. Implement the applications of algorithms for sorting, pattern matching etc

### UNIT – I:

**Introduction to Data Structures:** Introduction to the Theory of Data Structures, Data Representation, Abstract Data Types, Data Types, Primitive Data Types, Data Structure and Structured Type, Atomic Type, Difference between Abstract Data Types, Data Types, and Data Structures.

**Principles of Programming and Analysis of Algorithms:** Algorithms, Complexity, Big ‘O’ Notation, Algorithm Analysis, Structured Approach to Programming, Recursion.

### UNIT – II:

**Arrays:** Introduction to Linear and Non- Linear Data Structures, One- Dimensional Arrays, Array Operations, Two- Dimensional arrays, Multidimensional Arrays, Pointers and Arrays, an Overview of Pointers.

**Linked Lists:** Introduction to Lists and Linked Lists, Dynamic Memory Allocation, Basic Linked List Operations, Doubly Linked List, Circular Linked List, Atomic Linked List, Linked List in Arrays, Linked List versus Arrays

### UNIT – III:

**Stacks:** Introduction to Stacks, Stack as an Abstract Data Type, Representation of Stacks through Arrays, Representation of Stacks through Linked Lists, Applications of Stacks, Stacks and Recursion

**Queues:** Introduction, Queue as an Abstract data Type, Representation of Queues through Arrays, Representation of Queues through Linked Lists, Circular Queues, Double Ended Queues-Dequeues, Priority Queues, Application of Queues

### UNIT – IV:

**Binary Trees:** Introduction to Non- Linear Data Structures, Introduction Binary Trees, Types of Trees, Basic Definition of Binary Trees, Properties of Binary Trees, Representation of Binary

**B.Sc (Computer Science) Syllabus**  
**Rayalaseema University, Kurnool with effect from 2020-2021**

Trees, Operations on a Binary Search Tree, Binary Tree Traversal, Counting Number of Binary Trees, Applications of Binary Tree

**UNIT – V:**

**Searching and sorting:** Sorting – An Introduction, Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Sort. Searching – An Introduction, Linear or Sequential Search, Binary Search.

**Graphs:** Introduction to Graphs, Terms Associated with Graphs, Sequential Representation of Graphs, Linked Representation of Graphs, Traversal of Graphs, Spanning Trees, Shortest Path, Application of Graphs.

**BOOKS:**

1. “Data Structures using C”, ISRD group Second Edition, TMH
2. “Data Structures through C”, Yashavant Kanetkar, BPB Publications
3. “Data Structures Using C” Balagurusamy E. TMH

**RECOMMENDED CO-CURRICULAR ACTIVITIES:**

(Co-curricular activities shall not promote copying from textbook or from others work and shall encourage self/independent and group learning)

**A. Measurable**

1. Assignments (in writing and doing forms on the aspects of syllabus content and outside the syllabus content. Shall be individual and challenging)
2. Student seminars (on topics of the syllabus and related aspects (individual activity))
3. Quiz (on topics where the content can be compiled by smaller aspects and data (Individuals or groups as teams))
4. Study projects (by very small groups of students on selected local real-time problems pertaining to syllabus or related areas. The individual participation and contribution of students shall be ensured (team activity))

**B. General**

1. Group Discussion
2. Others

**RECOMMENDED CONTINUOUS ASSESSMENT METHODS:**

Some of the following suggested assessment methodologies could be adopted;

1. The oral and written examinations (Scheduled and surprise tests),
2. Closed-book and open-book tests,
3. Programming exercises,
4. Practical assignments and laboratory reports,
5. Observation of practical skills,
6. Individual and group project reports.
7. Efficient delivery using seminar presentations,
8. Viva voce interviews.
9. Computerized adaptive testing, literature surveys and evaluations,
10. Peers and self-assessment, outputs form individual and collaborative work