

Course Title		Data Structures
Course Code	CE307	
Course Credit	Theory	:03
	Practical	:01
	Tutorial	:00
	Credits	:04
Course Learning Outcomes		
<p>On the completion of the course, students will be able to :</p> <ul style="list-style-type: none"> <li>• <b>Understand</b> the characteristics of linear and non-linear data structures.</li> <li>• <b>Write</b> programs using various data structures.</li> <li>• <b>Analyze</b> different sorting and searching algorithms.</li> <li>• <b>Demonstrate</b> hashing and hashing methods.</li> </ul>		
Detailed Syllabus		
Sr. No.	Name of chapter & details	Hours Allotted
<b>Section – I</b>		
1	<b>Review of user defined function, Recursion, pointer, structure</b>	<b>05</b>
2	<b>Introduction to Data Structures and stack</b> Primitive & Non-primitive data structure, introduction of stack, basic operations (PUSH, POP, PEEK, CHANGE) on stack with example and major steps, Stack as Abstract Data Types and examples, Applications of Stacks-infix to postfix conversion and evaluation of postfix expression	<b>06</b>
3	<b>Queues :</b> Basic Operations (INSERT, DELETE) on queue, types of queue (circular queue, d-queue, priority queue) ,Applications: Simulation, Priority Queues.	<b>05</b>
4	<b>Linked list :</b> Overview of linked list, Dynamic Memory Allocation Linked List Operations (insert first, insert last, insert at any position, delete first, delete last, delete any node, search node, traversal, copy linked list, merge linked list). Types of linked list (Singly, Circular, Doubly, Sorted ) Applications of Linked Lists.	<b>08</b>
<b>Section – II</b>		

<b>5</b>	<b>Sorting &amp; Searching</b> Algorithm performance measurement, Introduction to Sorting, Selection Sort, Bubble Sort, Insertion Sort, Shell Sort, Merge Sort, Heap Sort, Quick Sort, Radix Sort, Sequential Search , Binary Search , comparison of sorting algorithm	<b>08</b>
<b>6</b>	<b>Nonlinear Data Structures:</b> Introduction to tree, Representations of Trees , Operations on binary tree & binary search tree (insert, traversal, search, delete) Conversion of General Tree to Binary Trees , Expression Trees, multi-way trees, threaded tree, balanced binary trees	<b>10</b>
<b>7</b>	<b>Hashing and collision:</b> Introduction, has table, hash function, types of hash function, collisions, collisions resolution	<b>04</b>
<b>8</b>	<b>GRAPHS</b> Introduction to graph, Representation of Graphs	<b>02</b>

**Instructional Method and Pedagogy**

- Lectures will be conducted with the aid of multi-media projector, black board, or model
- Interactive exchange with students in class.
- Assignments based on course content will be given to the students at the end of each unit/topic and will be evaluated at regular interval.
- The course includes tutorials, where students have an opportunity to practice the examples for the concepts being taught in lectures.
- Discussion sessions with optional additional lectures and labs with group discussion.

**Reference Books**

- Reema Thareja, Data Structure using C, Oxford University press, Third edition, 2012
- Y.Kanitkar, Data Structure Using C, Second Edition, PHI
- G S Baluja ,Data Structure through C, 2009, Dhanpat Rai & Co
- Aron M. Tenenbaum, Data structure using C and C++, PHI

### Additional Resources

- NPTEL Video Lectures of ...
  - 1) Data structures and Algorithms of Computer Science & Engineering by Prof. Naveen Garg, IIT Delhi [Available at: <http://nptel.ac.in/courses/106102064/>]
  - 2) Programming, Data Structures and Algorithms by Prof. Hema A. Murthy, IIT Madras [Available at: <http://nptel.ac.in/courses/106106127/2>]
  - 3) Data Structures and Program Methodology by Prof. Pradeep K. Das, IIT Guwahati [Available at: <http://nptel.ac.in/courses/106103069/>]

### List of Experiments

1. **Write** C programs for the followings:
  - I. **Implement** basic arithmetic calculator with and without function.
  - II. **Create** a array of n element. **Find** min & max value. Also **compute** average value (Note:- with and without function ).
  - III. **Design** a structure of student with members i.e. roll\_no and name and marks of 3 subjects. **Use** this structure to **find** highest marks of each subject, ranker in a class of 70 students.
2. **Write** C programs for the followings:
  - I. **Design** a data structure “stack” with necessary members(array,top,size). Also implement operations on stack (push, pop, peep, change). (Note:- with and without function ).
  - II. **Design** a data structure “stack” using “struct” with necessary members(array,top,size). Also implement operations on stack (push, pop, peep, change). (Note:- with and without function ).
  - III. **Develop** the following stack application
    - a. Infix to postfix conversion
    - b. **Evaluation** of postfix expression
    - c. To **demonstrate** the **use** of multiple stack
3. **Write** C programs for the followings using “struct”:
  - I. **Design** a data structure “queue” with necessary members. Also implement operations on queue (insert and delete).
  - II. **Develop** a data structure “circular queue” with necessary members. Also implement operations on queue (insert and delete).
  - III. **Construct** a data structure “double ended queue” with necessary members. Also implement operations on queue (insert and delete).
  - IV. **Create** a data structure “priority queue” with necessary members. Also implement operations on queue (insert and delete).
4. **Write** C programs for the followings using D.M.A
  - I. **Design** a data structure singly linked list with necessary members. Also implement operations on linked list (insert at first, insert at last, insert in between and delete from first, last and between, search, display).
  - II. **Develop** a data structure circular singly linked list with necessary members. Also implement operations on circular linked list (insert at first, insert at last, insert in between and delete from first, last and between, search, display).
  - III. **Demonstrate** a data structure doubly linked list with necessary members. Also implement operations on doubly linked list (insert at first, insert at last, insert in between and delete from first, last and between, search, display).

5. Write C programs for the following
- I. **Apply** selection and bubble sort method to arrange the given below data in ascending order  
12,54,23,6,78,89,5,7,34
  - II. **Apply** insertion and shell sort method to arrange the given below data in ascending order  
11,55,22,6,78,89,55,77,34
  - III. **Analyze** merge and quick sort method to arrange the given below data in ascending order  
10,51,21,63,7,9,53,77,33
  - IV. **Apply** radix sort method to arrange the given below data in ascending order  
12,54,23,6,78,89,5,7,34
  - V. **Differentiate** Linear and binary search on the following array based on its implementation.  
5 ,6,7,10,15,25,56,67,89,90
6. Write C programs for the following
- I. **Create** a binary search tree with at least 5 nodes.
  - II. **Construct** a binary search tree with N nodes. Also practice traversal operation (in order, pre order, post order).
  - III. **Develop** threaded binary tree.