

<b>DATA STRUCTURES LABORATORY</b> [As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2015 -2016) <b>SEMESTER - III</b>			
Laboratory Code	<b>15CSL38</b>	IA Marks	<b>20</b>
Number of Lecture Hours/Week	<b>01I + 02P</b>	Exam Marks	<b>80</b>
Total Number of Lecture Hours	<b>40</b>	Exam Hours	<b>03</b>
<b>CREDITS - 02</b>			
<b>Course objectives:</b> This laboratory course enable students to get practical experience in design, develop, implement, analyze and evaluation/testing of <ul style="list-style-type: none"> <li>• Asymptotic performance of algorithms.</li> <li>• Linear data structures and their applications such as Stacks, Queues and Lists</li> <li>• Non-Linear Data Structures and their Applications such as Trees and Graphs</li> <li>• Sorting and Searching Algorithms</li> </ul>			
<b>Descriptions (if any)</b>  <b>Implement all the experiments in C Language under Linux / Windows environment.</b>			
<b>Laboratory Experiments:</b> <ol style="list-style-type: none"> <li>Design, Develop and Implement a menu driven Program in C for the following <b>Array</b> operations <ol style="list-style-type: none"> <li>Creating an Array of <b>N</b> Integer Elements</li> <li>Display of Array Elements with Suitable Headings</li> <li>Inserting an Element (<b>ELEM</b>) at a given valid Position (<b>POS</b>)</li> <li>Deleting an Element at a given valid Position(<b>POS</b>)</li> <li>Exit.</li> </ol> Support the program with functions for each of the above operations. </li> <li>Design, Develop and Implement a Program in C for the following operations on <b>Strings</b> <ol style="list-style-type: none"> <li>Read a main String (<b>STR</b>), a Pattern String (<b>PAT</b>) and a Replace String (<b>REP</b>)</li> <li>Perform Pattern Matching Operation: Find and Replace all occurrences of <b>PAT</b> in <b>STR</b> with <b>REP</b> if <b>PAT</b> exists in <b>STR</b>. Report suitable messages in case <b>PAT</b> does not exist in <b>STR</b></li> </ol> Support the program with functions for each of the above operations. Don't use Built-in functions. </li> <li>Design, Develop and Implement a menu driven Program in C for the following operations on <b>STACK</b> of Integers (Array Implementation of Stack with maximum size <b>MAX</b>) <ol style="list-style-type: none"> <li><b>Push</b> an Element on to Stack</li> <li><b>Pop</b> an Element from Stack</li> <li>Demonstrate how Stack can be used to check <b>Palindrome</b></li> <li>Demonstrate <b>Overflow</b> and <b>Underflow</b> situations on Stack</li> </ol> </li> </ol>			

- e. Display the status of Stack
- f. Exit

Support the program with appropriate functions for each of the above operations

4. Design, Develop and Implement a Program in C for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, \*, /, %(Remainder), ^(Power) and alphanumeric operands.
5. Design, Develop and Implement a Program in C for the following Stack Applications
  - a. Evaluation of **Suffix expression** with single digit operands and operators: +, -, \*, /, %, ^
  - b. Solving **Tower of Hanoi** problem with **n** disks
6. Design, Develop and Implement a menu driven Program in C for the following operations on **Circular QUEUE** of Characters (Array Implementation of Queue with maximum size **MAX**)
  - a. Insert an Element on to Circular QUEUE
  - b. Delete an Element from Circular QUEUE
  - c. Demonstrate **Overflow** and **Underflow** situations on Circular QUEUE
  - d. Display the status of Circular QUEUE
  - e. Exit

Support the program with appropriate functions for each of the above operations

**Continued:**

7. Design, Develop and Implement a menu driven Program in C for the following operations on **Singly Linked List (SLL)** of Student Data with the fields: **USN, Name, Branch, Sem, PhNo**
  - a. Create a **SLL** of **N** Students Data by using **front insertion**.
  - b. Display the status of **SLL** and count the number of nodes in it
  - c. Perform Insertion / Deletion at End of **SLL**
  - d. Perform Insertion / Deletion at Front of **SLL(Demonstration of stack)**
  - e. Exit
8. Design, Develop and Implement a menu driven Program in C for the following operations on **Doubly Linked List (DLL)** of Employee Data with the fields: **SSN, Name, Dept, Designation, Sal, PhNo**
  - a. Create a **DLL** of **N** Employees Data by using **end insertion**.
  - b. Display the status of **DLL** and count the number of nodes in it
  - c. Perform Insertion and Deletion at End of **DLL**
  - d. Perform Insertion and Deletion at Front of **DLL**
  - e. Demonstrate how this **DLL** can be used as **Double Ended Queue**
  - f. Exit

<p>9. Design, Develop and Implement a Program in C for the following operations on <b>Singly Circular Linked List (SCLL)</b> with header nodes</p> <ol style="list-style-type: none"> <li>Represent and Evaluate a Polynomial <math>P(x,y,z) = 6x^2y^2z - 4yz^5 + 3x^3yz + 2xy^5z - 2xyz^3</math></li> <li>Find the sum of two polynomials <b>POLY1(x,y,z)</b> and <b>POLY2(x,y,z)</b> and store the result in <b>POLYSUM(x,y,z)</b></li> </ol> <p>Support the program with appropriate functions for each of the above operations</p> <p>10. Design, Develop and Implement a menu driven Program in C for the following operations on <b>Binary Search Tree (BST)</b> of Integers</p> <ol style="list-style-type: none"> <li>Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2</li> <li>Traverse the BST in Inorder, Preorder and Post Order</li> <li>Search the BST for a given element (<b>KEY</b>) and report the appropriate message</li> <li>Exit</li> </ol> <p>11. Design, Develop and Implement a Program in C for the following operations on <b>Graph(G)</b> of Cities</p> <ol style="list-style-type: none"> <li>Create a Graph of N cities using Adjacency Matrix.</li> <li>Print all the nodes <b>reachable</b> from a given starting node in a digraph using DFS/BFS method</li> </ol> <p>12. Given a File of N employee records with a set K of Keys(4-digit) which uniquely determine the records in file F. Assume that file F is maintained in memory by a Hash Table(HT) of m memory locations with L as the set of memory addresses (2-digit) of locations in HT. Let the keys in K and addresses in L are Integers. Design and develop a Program in C that uses Hash function <b>H: K → L</b> as <math>H(K) = K \bmod m</math> (<b>remainder</b> method), and implement hashing technique to map a given key K to the address space L. Resolve the collision (if any) using <b>linear probing</b>.</p>	
<p><b>Course outcomes:</b></p> <p>On the completion of this laboratory course, the students will be able to:</p> <ul style="list-style-type: none"> <li>Analyze and Compare various linear and non-linear data structures</li> <li>Code, debug and demonstrate the working nature of different types of data structures and their applications</li> <li>Implement, analyze and evaluate the searching and sorting algorithms</li> <li>Choose the appropriate data structure for solving real world problems</li> </ul>	
<p><b>Graduate Attributes (as per NBA)</b></p> <ol style="list-style-type: none"> <li>Engineering Knowledge</li> <li>Problem Analysis</li> <li>Design/Development of Solutions</li> <li>Modern Tool Usage</li> </ol>	
<p><b>Conduction of Practical Examination:</b></p> <ol style="list-style-type: none"> <li>All laboratory experiments (<b>TWELVE</b> nos) are to be included for practical examination.</li> <li>Students are allowed to pick one experiment from the lot.</li> <li>Strictly follow the instructions as printed on the cover page of answer script</li> <li>Marks distribution: Procedure + Conduction + Viva: <b>20 + 50 + 10 (80)</b></li> <li><b>Change of experiment is allowed only once and marks allotted to the procedure part to be made zero.</b></li> </ol>	