



# MANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL

(A constituent unit of MAHE, Manipal)

## COURSE PLAN

Department	:	Computer Applications			
Course Name & code	:	Data Structures Lab & DSE-2163			
Semester & branch	:	III Semester B.Tech & DSE			
Name of the faculty	:	Mrs. Linda Varghese, Dr. Savitha G			
No of contact hours/week:		L	T	P	C
		0	0	3	1

## ASSESSMENT PLAN

### Course Outcomes (COs)

<i>At the end of this course, the student should be able to:</i>		<b>No. of Contact Hours</b>	<b>Marks</b>
CO1:	Identify appropriate data structure	3	18
CO2:	Interpret the working of searching and sorting techniques	3	18
CO3:	Demonstrate the working of linear and non-linear data structure	12	46
CO4:	Make use of preliminary structures to implement various applications	6	18
<b>Total</b>		24	100

<b>1. Continuous Evaluation</b>	60%
---------------------------------	-----

Total of 3 regular evaluations which will be carried out in alternate weeks. Each evaluation is for 20 marks of which will have the following split up: Record: 4 Marks; Viva: 10 Marks; Execution: 6 Marks; Total = 20 Marks  
Total Internal Marks: 3 \* 20 =60 Marks

## 2. Lab Examination

40%

- Examination of 2 hours duration (Max. Marks: 40). Program Write up: 15 Marks; Program Execution: 25; Marks Total: 15+25 =40 Marks

## Course Plan

L. No.	Topics	Course Outcome Addressed
L1	<b>1. 1. Implementation of Stack using arrays</b> <b>2. To check if the given parenthesized expression has properly matching open and closing parenthesis</b> <b>3. To check a given string is palindrome or not using stack</b> <b>4. Tower of Hanoi for n disks(Recursion application)</b>  <b>Additional questions:</b>  <b>1.) Solving problems using Recursion:</b> a) Binary search b) Factorial of a given number c) GCD of 2 numbers d) Fibonacci series upto nth term  <b>2.) Write a program to find the transpose of a sparse matrix represented using array of objects.</b>	CO1
L2	<b>1. Conversion of infix expression to postfix and prefix forms</b> <b>2. Evaluation of postfix and prefix expressions</b>  <b>Additional Questions:</b>  <b>1.) Write a program to perform following string operations without using string handling functions:</b> a) length of the string b) string concatenation c) string comparison d) to insert a sub string e) to delete a substring  <b>2.) Mapping of 2-D arrays to 1-D arrays:</b> <b>Map the following 2-D arrays (matrices) to 1-D arrays (lists).</b>	CO4

	<p>a) Upper triangular matrix b) Lower triangular matrix c) Diagonal matrix d) Tri-diagonal matrix e) Row-major f) Column-major</p> <p>Display the element at any specified position (row, column).</p> <p>3.) Define a class time with data members hour, min, sec .Write the user defined functions to (i) Add (ii) To find difference between two objects of class time. Functions take two time objects as argument and return time object. Also write the display and read function.</p>									
L3	<p>1. Implementation of Queue using arrays 2. Implement a circular queue of Strings with functions insert, delete and display. 3. Write a program to implement the circular queue using arrays</p> <p><b>Additional Questions:</b></p> <p>1.) Define a class EMPLOYEE contains following members. Data members: Employee_Number, Employee_Name, Basic, DA, IT, Net_Sal, Gross_salary. Member functions: To read the data, calculate net salary, gross salary and display both salary. Write a C++ program to read the data of N employees and compute Net salary and Gross salary of each employee. (DA= 12% of Basic and Income Tax (IT) = 18% of the gross salary).</p> <p>2.) Create a flight class that has private data members: flight number (integer), destination(characters), distance (float), fuel (float).</p> <p>a) Initialize fuel to 13.2 liters b) Provide a parameterized function that accepts fuel details c) Private Member functions: calculate_fuel() to calculate the value of Fuel as per the following criteria:</p> <table><tr><td>Distance (in kilometers)</td><td>Fuel (in liters)</td></tr><tr><td>&lt;=1000</td><td>500</td></tr><tr><td>&gt;1000 and &lt;=2000</td><td>1100</td></tr><tr><td>&gt;2000</td><td>2200</td></tr></table> <p>d) Member functions: information_entry() to allow user to enter values for flight number, destination, distance which calls function calculate_fuel() to calculate the quantity of fuel and display_info() to allow user to view flight details.</p>	Distance (in kilometers)	Fuel (in liters)	<=1000	500	>1000 and <=2000	1100	>2000	2200	CO1
Distance (in kilometers)	Fuel (in liters)									
<=1000	500									
>1000 and <=2000	1100									
>2000	2200									

	<p>3.) Check if a given number is prime or not</p> <p>4.) Factorial of given 10 numbers (do not use arrays)</p> <p>5.) Print all odd numbers between m and n</p> <p>6.) Menu driven program to sum all elements entered upto -1</p> <p>7.) Find <math>\sin(x)</math> using series</p> <p>8.) Find <math>\cos(x)</math> using series</p> <p>9.) Find <math>e^x</math> using series</p> <p>10.) Print triangle in the following form using loops until n.</p> <p>Ex. If <math>n=6</math></p> <pre> 1 2   3 4   5   6 </pre>	
L4	<p>1. Write a menu driven program to perform the following operations on linked list.</p> <ol style="list-style-type: none"> <li>Insert an element in the beginning of the list</li> <li>Insert an element at the end of the list</li> <li>Insert an element before another element in the existing list</li> <li>Insert an element after another element in the existing list</li> <li>Delete a given element from the list</li> <li>Print the list</li> </ol> <p>2. Implement Stack and Queue using linked lists</p> <p><b>Additional Questions:</b></p> <p>1.) Write the program which performs the following functions:</p> <ol style="list-style-type: none"> <li>Reverse the list</li> <li>Sort the list</li> <li>Delete every alternate node in the list</li> <li>Insert an element in a sorted list such that the order is maintained</li> </ol> <p>2.) Representation of a Sparse Matrix:- Represent a sparse matrix using 1-D array. Use this 1-D array to reconstruct the original matrix.</p> <p>3.) Representation of a Polynomial:- Represent a polynomial using 1-D array and perform addition operation on two polynomials.</p>	CO3
L5	<p>1. Write a menu driven program to perform the following on a doubly linked list</p> <ol style="list-style-type: none"> <li>Insert an element at the rear end of the list</li> <li>Delete an element from the rear end of the list</li> <li>Insert an element at a given position of the list</li> </ol>	CO3

	<p>d) Delete an element from a given position of the list  e) Insert an element after another element  f) Insert an element before another element  g) Print the list</p> <p>2. Write a program to add two polynomials using doubly linked list.</p> <p>Additional Questions:</p> <p>1.) Write a program to add two polynomials using doubly linked list.  2.) Write a program to add 2 polynomials using circular doubly linked list with head node.</p>	
L6	<p>1. Write user defined functions to perform the following operations on binary trees:</p> <p>a) Iteratively create a binary tree  b) In order traversal (Iterative)  c) Post order traversal (Iterative)  d) Preorder traversal (Iterative)  e) Count the number of leaf nodes in a binary tree</p> <p>2. Write a program to perform the following:</p> <p>a) Print the parent of the given element  b) Print the depth of a tree  c) Print the ancestors of a given node</p> <p>3. Write a program to search for a given element in a binary search tree.</p> <p>Additional Questions:</p> <p>1.) Write a program to implement level order traversal on binary search tree  2.) Write a program to insert an element into a binary search tree.  3.) Write a program to delete an element from a binary search tree.</p>	CO3
L7	<p>1. Linear Search and Binary searching  2. Sorting: Bubble, Quick, Selection &amp; Insertion</p> <p>Additional Questions:</p> <p>1.) Heap sort  2.) Radix sort  3.) Merge sort</p>	CO2

<b>L8</b>	<b>End Semester Examination</b>	

### References:

1. Ellis Horowitz, Sartaj Sahni, Dinesh Mehta, Fundamentals of Data Structures in C++, 2nd Edition, Galgotia Publications, Reprint 2013
2. BEHROUZ A. FOROUZAN, RICHARD F. GILBERG, A STRUCTURED PROGRAMMING APPROACH USING C, 3E, CENGAGE, LEARNING INDIA PVT.LTD, INDIA, 2007.
3. Behrouz A. Forouzan, Richard F. Gilberg, Data Structures, A Pseudocode approach Using C, 2e, Cengage, learning India Pvt.Ltd, India, 2009.
4. Ellis Horowitz, Sartaj Sahni, Dinesh Mehta, Fundamentals of Data Structures in C, 2nd Edition Universities Press (India) Private Limited, Reprint, 2013
5. Debasis Samanta, Classic Data structures- 2nd edition, PHI Learning Private Limited , 2010
6. Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, 2nd Edition, Pearson Education, 2005.
7. Michael T, Goodrich, Roberto Tamassia, David Mount, Data Structures and Algorithms in C++, 2nd Edition, John Wiley & Sons, 2011

**Submitted by:** Linda Varghese and Dr. Savitha G

**(Signature of the faculty)**

**Date:** 18-10-2021

**Approved by:** Dr. Karunakar A Kotegar

**(Signature of HOD)**

**Date:** 18-10-2021

**FACULTY MEMBERS TEACHING THE COURSE (IF MULTIPLE SECTIONS EXIST):**

<b>FACULTY</b>	<b>SECTION</b>	<b>FACULTY</b>	<b>SECTION</b>
Mrs. Linda Varghese	A		
Dr. Savitha G	B		

\*\*\*\*\*