Course Code CSE2002	Data Structures and Algorithms	CT C LTP 4
Prerequisite:	Introduction to Problem Solving and Programming	

Objectives:

1. To understand various types of fundamental data structures (standard and user defined).
2. To learn about algorithm analysis for the run time complexities and the space requirements.
3. To acquire knowledge of data structures and algorithms for implementing various real-world problems.

Expected Outcomes:

Students will be able to

1. Apply the fundamental knowledge of various data structures and algorithms to analyze, design, formulate and implement algorithm for any real time problem.
2. Apply current techniques in data structures and algorithmic principles for modeling and developing software systems.
3. Choose an appropriate design paradigm that solves the given problem efficiently along with appropriate data structures.
4. Map real-world problems to algorithmic solutions.

Student Outcomes (SO): a,b,i,l,m

a. An ability to apply the knowledge of mathematics, science and computing appropriate to the discipline

b. An ability to analyze a problem, identify and define the computing requirements appropriate to its solution.

i. Design and conduct experiments as well as analyze and interpret data

l. An ability to apply mathematical foundations, algorithmic principles and computer science theory in the modelling and design of computer-based systems (CS)

m. An ability to apply design and development principles in the construction of software systems (CS).

Unit No	Unit Content	No. of hours	SOs
1	**Introduction to Algorithm and Data Structures** Algorithm: Introduction - Algorithm Design – Complexity- Asymptotic notations. Data Structures: Introduction- Classification of Data structure - Abstract Data Type (ADT).	**6 Hours**	a,b
2	**Sorting and Searching** Brute force approach: General method -Sorting (bubble, selection, insertion) –Searching (Sequential/Linear) Divide and Conquer approach: General method - Sorting (merge, quick) – Searching (Binary Search).	**8 Hours**	a,b,l
3	**List, Statck and Queue ADT** Linked List: Array Vs Linked List - Singly Linked List, Doubly Linked Lists – Circular Linked Lists-implementation - application. Stack and Queue: Introduction – implementation (static and dynamic) – application – Circular queues-application.	**10 Hours**	a,b,l,m

4	**TREES AND HASHING** Linear Vs Non-Linear Data Structures -General Tree – Terminologies -Binary Tree – Expression Tree - Traversals - Binary Search Tree – AVL Tree – Red block Tree – Splay Tree – B Tree. - Hashing: Introduction – Hash Function-Methods-Collision Resolution.	**10 Hours**	a,b,l,m
5	**Graph ADT** Graph: Introduction – Representations – Traversals - Topological Sorting – Connected and Bi-Connected Components – Articulation Point - Shortest-path algorithms (Dijkstra's and Floyd's algorithms) - Minimum spanning tree (Prim's and Kruskal's algorithms).	**9 Hours**	a,b,l,m
	Guest Lecture on Contemporary Topics	**2 Hours**	
	Total Lecture:	**45 Hours**	

Mode of Teaching and Learning: Flipped Class Room, One Lecture to be videotaped, Digital/Computer based models to augment lecture for practice/tutorial, 2 hours lectures by industry experts on contemporary topics

Mode of Evaluation and assessment:

The assessment and evaluation components may consist of unannounced open book examinations, quizzes, student's portfolio generation and assessment, and any other innovative assessment practices followed by faculty, in addition to the Continuous Assessment Tests and Final Examinations.

Text Books:

1.	Thomas H. Cormen , Charles E. Leiserson , Ronald L. Rivest , Clifford Stein, Introduction to Algorithms, 3rd Edition, MIT Press, 2009

Reference Books:

1.	Mark A. Weiss,Data Structures and Algorithm Analysis in C++, 4th Edition, Pearson, 2014
2.	Aaron M. Tenenbaum, Yeedidyah Langsam, Moshe J. Augenstein, 'Data structures using C', Pearson Education, 2010.
3.	D. E. Knuth, Art of computer programming, Volume 1: Fundamental algorithms, Addison Wesley, 2011.

Recommendation by the Board of Studies on	
Approval by Academic council on	
Compiled by	

Indicative List of Experiments

No.	Description	SO
1	In the situation where there are multiple users or a network computer system, you probably share a printer with other users. When you request to print a file, your request is added to the print queue. When your request reaches the front of the print queue, your file is printed. This ensures that only one person at a time has access to	

	the printer and that this access is given on a first-come, first- served basis. Design an algorithm for this scenario and implement your algorithm in any programming language .	
2	Implement an effective solution for Balanced parenthesis problem.	
3	You are making an iPod playlist to hear the songs. Assuming that shuffle functions are not applicable, choose an appropriate data structure that will add and delete songs onto you're your iPod in such a way that the recently inserted song will always be the first song currently on the iPod.	
4	You have n coins, all of which are gold except one coin which appears to be a gold coin, but it is fake. All gold coins are of the same weight, the fake coin weighs less than the others. You have a balance scale, you can put any number of coins on each side of the scale at one time and it will tell you if the two sides weight the same, or which side is lighter if they don't weight the same. The problem is to identify the fake coin. Design an algorithm to find the fake coin in the given n coins.	
5	Implement the following operations on the stack using linked list data structure. In addition to the usual operations of the stack and the linked list, your implementation should handle two more operations. o Split(p(i1,i2,i3…ip),q(j1,j2,…jq)) in the stack, where the stack of length n will be split in two stacks, each of length p and q such that p+q=n. Here the index i_k+1 need not be equalto $i_{k+1,}$ where 1≤k≤p and the index j_i+1 need not be equalto $j_{i+1,}$ where 1≤i≤q. o Given two stacks p and q , Combine((p(i1,i2,i3…ip),q(j1,j2,…jq)) into one stack of length p+q=n. The new stack should contain the elements of the stacks p and q in any combination.	a,b,i,l,m
6	Consider the equation APPLE + LEMON = BANANA. Assume that each letter actually represents a digit from 0 to 9. Some conditions are imposed. The leftmost letter can't be zero in any word. There must be a one-to-one mapping between letters and digits. In other words, if you choose the digit 5 for the letter E, then all of the E's in the equation must be 5 and no other letter can be a 5. No digit can be repeated.	
7	Given a linked list and a number k. Write an algorithm to reverse every k nodes in the list. **Example :** Input: 1→2→3→4→5→6 and k = 3 Output: 3→2→1→6→5→4	
8	Construct a binary search tree, and perform the following operations. (i)Inserting an element. (ii) Search for an element (iii) traversal.	

9	Given two values k1 and k2 (where k1 < k2) and a root pointer to a Binary Search Tree. Print all the keys of tree in range k1 to k2. i.e. print all x such that k1<=x<=k2 and x is an element of given BST.	
10	Design an algorithm to find the shortest path from source vertex to all the other vertices, in a graph. Apply your algorithm to the below graph to find the shortest path, assume the source vertex is '1'.