DESIGN AND ANALYSIS OF ALGORITHM LABORATORY
[As per Choice Based Credit System (CBCS) scheme]
SEMESTER - IV

Subject Code:	**18CSL47**	IA Marks:	40
Number of Lecture Hours/Week:	01 I + 02 P	Exam Marks:	60
Total Number of Lecture Hours:	40	Exam Hours:	03

CREDITS - 02

Course objectives: This course will enable students to

- Design and implement various algorithms in JAVA.
- Employ various design strategies for problem solving.
- Measure and compare the performance of different algorithms.

Example programs are given from page 3 – 18 and viva questions are attached at last.

Exp No.	Part	Description	Page No.
1	A	Create a Java class called ***Student*** *with* the following details as variables within it. (i) USN (ii) Name (iii) Branch (iv) Phone Write a Java program to create ***n Student*** objects and print the USN, Name, Branch, and Phone of these objects with suitable headings.	19
	B	Write a Java program to implement the Stack using arrays. Write Push(), Pop(), and Display() methods to demonstrate its working.	21
2	A	Design a superclass called ***Staff*** with details as StaffId, Name, Phone, Salary. Extend this class by writing three subclasses namely ***Teaching*** (domain, publications), ***Technical*** (skills), and ***Contract*** (period). Write a Java program to read and display at least 3 ***staff*** objects of all three categories.	24
	B	Write a Java class called ***Customer*** to store their name and date_of_birth. The date_of_birth format should be dd/mm/yyyy. Write methods to read customer data as <name, dd/mm/yyyy> and display as <name, dd, mm, yyyy> using StringTokenizer class considering the delimiter character as "/".	28
3	A	Write a Java program to read two integers ***a*** and ***b***. Compute ***a/b*** and print, when ***b*** is not zero. Raise an exception when ***b*** is equal to zero.	29
	B	Write a Java program that implements a multi-thread application that has three threads. First thread generates a random integer for every 1 second; second thread computes the square of the number and prints; third thread will print the value of cube of the number.	30
4		Sort a given set of ***n*** integer elements using Quick Sort method and compute its time complexity. Run the program for varied values of *n*>5000 and record the time taken to sort. Plot a graph of the time taken versus ***n*** on graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate using Java how the divide- and-conquer method works along with its time complexity analysis: worst case, average case and best case.	32

5	Sort a given set of ***n*** integer elements using **Merge Sort** method and compute its time complexity. Run the program for varied values of *n*>5000, and record the time taken to sort. Plot a graph of the time taken versus ***n*** on graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate using Java how the divide- and-conquer method works along with its time complexity analysis: worst case, average case and best case.	38
6	Implement in Java, the **0/1 Knapsack** problem using (a) Dynamic Programming method (b) Greedy method.	44
7	From a given vertex in a weighted connected graph, find shortest paths to other vertices using **Dijkstra's algorithm**. Write the program in Java.	50
8	Find Minimum Cost Spanning Tree of a given connected undirected graph using **Kruskal's algorithm.** Use Union-Find algorithms in your program.	54
9	Find Minimum Cost Spanning Tree of a given connected undirected graph using **Prim's algorithm**.	57
10	Write Java programs to (a) Implement All-Pairs Shortest Paths problem using **Floyd's algorithm**. (b) Implement **Travelling Sales Person problem** using Dynamic programming.	61 63
11	Design and implement in Java to find a **subset** of a given set **S** = {Sl, S2 ,... ,Sn} of ***n*** positive integers whose SUM is equal to a given positive integer ***d.*** For example, if S = {1, 2, 5, 6, 8} and *d*= 9, there are two solutions {1,2,6}and {1,8}. Display a suitable message, if the given problem instance doesn't have a solution.	65
12	Design and implement in Java to find all **Hamiltonian Cycles** in a connected undirected Graph G of *n* vertices using backtracking principle.	68

Course Outcomes: The students should be able to:

- Design algorithms using appropriate design techniques (brute-force, greedy, dynamic programming, etc.)
- Implement a variety of algorithms such assorting, graph related, combinatorial, etc., in a high level language.
- Analyze and compare the performance of algorithms using language features.
- Apply and implement learned algorithm design techniques and data structures to solve real- world problems.

Graduate Attributes

- Engineering Knowledge
- Problem Analysis
- Modern Tool Usage
- Conduct Investigations of Complex Problems
- Design/Development of Solutions

Conduction of Practical Examination:

All laboratory experiments (Twelve problems) are to be included for practical examination. Students are allowed to pick one experiment from the lot.

To generate the data set use random number generator function.

Strictly follow the instructions as printed on the cover page of answer script for breakup of marks

Marks distribution: Procedure + Conduction + Viva: 15 + 70 + 15 (100). Change of experiment is allowed only once and marks allotted to the procedure