

Exp No.	Part	Description
1	A	<p>Create a Java class called <b>Student</b> with the following details as variables within it.</p> <ul style="list-style-type: none"> <li>(i) USN</li> <li>(ii) Name</li> <li>(iii) Branch</li> <li>(iv) Phone</li> </ul> <p>Write a Java program to create <math>n</math> <b>Student</b> objects and print the USN, Name, Branch, and Phone of these objects with suitable headings.</p>
	B	Write a Java program to implement the Stack using arrays. Write Push(), Pop(), and Display() methods to demonstrate its working.
2	A	Design a superclass called <b>Staff</b> with details as StaffId, Name, Phone, Salary. Extend this class by writing three subclasses namely <b>Teaching</b> (domain, publications), <b>Technical</b> (skills), and <b>Contract</b> (period). Write a Java program to read and display at least 3 <b>staff</b> objects of all three categories.
	B	Write a Java class called <b>Customer</b> 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 "/".
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.
	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.
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.

5	Sort a given set of $n$ integer elements using <b>Merge Sort</b> 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.
6	Implement in Java, the <b>0/1 Knapsack</b> problem using (a) Dynamic Programming method (b) Greedy method.
7	From a given vertex in a weighted connected graph, find shortest paths to other vertices using <b>Dijkstra's algorithm</b> . Write the program in Java.
8	Find Minimum Cost Spanning Tree of a given connected undirected graph using <b>Kruskal's algorithm</b> . Use Union-Find algorithms in your program.
9	Find Minimum Cost Spanning Tree of a given connected undirected graph using <b>Prim's algorithm</b> .
10	Write Java programs to (a) Implement All-Pairs Shortest Paths problem using <b>Floyd's algorithm</b> . (b) Implement <b>Travelling Sales Person problem</b> using Dynamic programming.
11	Design and implement in Java to find a <b>subset</b> of a given set $S = \{S_1, S_2, \dots, S_n\}$ 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.
12	Design and implement in Java to find all <b>Hamiltonian Cycles</b> in a connected undirected Graph $G$ of $n$ vertices using backtracking principle.