

**DESIGN AND ANALYSIS OF ALGORITHMS LABORATORY**  
(Effective from the academic year 2018 -2019)

**SEMESTER – IV**

Course Code	18CSL47	CIE Marks	40
Number of Contact Hours/Week	0:2:2	SEE Marks	60
Total Number of Lab Contact Hours	36	Exam Hours	03

**Credits – 2**

**Course Learning Objectives:** This course (18CSL47) 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.

**Descriptions (if any):**

- Design, develop, and implement the specified algorithms for the following problems using Java language under LINUX /Windows environment. Netbeans / Eclipse or IntelliJIdea Community Edition IDE tool can be used for development and demonstration.
- **Installation procedure of the required software must be demonstrated, carried out in groups and documented in the journal.**

**Programs List:**

1.	
a.	Create a Java class called <i>Student</i> with the following details as variables within it. (i) USN (ii) Name (iii) Programme (iv) Phone Write a Java program to create <i>nStudent</i> objects and print the USN, Name, Programme, and Phone of these objects with suitable headings.
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 <i>Staff</i> with details as StaffId, Name, Phone, Salary. Extend this class by writing three subclasses namely <i>Teaching</i> (domain, publications), <i>Technical</i> (skills), and <i>Contract</i> (period). Write a Java program to read and display at least 3 <i>staff</i> objects of all three categories.
b.	Write a Java class called <i>Customer</i> 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 <i>a</i> and <i>b</i> . Compute <i>a/b</i> and print, when <i>b</i> is not zero. Raise an exception when <i>b</i> 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 <i>n</i> integer elements using <b>Quick Sort</b> method and compute its time complexity. Run the program for varied values of <i>n</i> > 5000 and record the time taken to sort. Plot a graph of the time taken versus <i>n</i> 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.

