

Advanced Java Programming Experiment List

1. Instructions

- Students/learners are advised to perform each experiments listed in the Table.
- All the experiments are compulsory to include in the lab records.
- Learners are advised to prepare a write-up according to the template given to you.
- You are advised to copy others work, if found zero marks will be given to you.
- For any difficulty you can come for help and discussion.
- You are advised to understand the instruction given by instructor/faculty during the lab session which will help you to achieve the goal.
- Your suggestions are always welcome.
- Marks will be awarded based on the content written in your words and your performance in oral session. You are advised to include references as appropriate.
- If any student fails to fulfill the requirements then he/she will be given second opportunity (2 days) to submit the revised version of the write-up. But, if student fail to submit within the time line given then zero marks will be awarded.

2. Experiment List

Lab No.	Experiment Details	Learning Outcome
1.	Write a program to demonstrate the working of multi threading. • Task 1: Use Threads class to enable thread. • Task 2: Write a program to create threads using runnable interface. you need to explain understanding and comparative study b/w thread class implementation and runnable interface for the programs you have created • Task 3: WAP to set priorities (min, max & norm) for the threads. Execute the program explain your understanding by looking the output.	At the end learners will be able to: • To describe the working of threads theoretically and experimentally. • Creating threads in Java programming language. • Setting the priority for threads.

2.	Create a RMI service to demonstrate the working of RMI • Task 1 : create an interface declare a function to find area of a rectangle • Task 2: Create a class to implement the interface created above. • Task 3: create a server and initialize the object of interface to invoke the methods. • Task 4: create the clients which will send request to the server.	At the end learners will be able to: • Describe the steps involved in creating RMI services. • Explain the beauty of using Interfaces. • Describe the working of UnicastRemoteObject. • Describe the working of Stub and Skeleton. • Explain the purpose of rmiregistry. • Explain the purpose of rmic.
3.	Creating a GUI for accepting the user id and user name. Use JDBC-ODBC bridge to connect the Java application developed with the database. Verify the connectivity and perform the following operations. • Task-1: Creating GUI with two labels ID and Name and use two text boxes to accept both. • Task-2: Use INSERT button. When user will click on it data which are given in both the text boxes must go into the Database. • Task-3: Use UPDATE button. When user want to update any information in the database then it must work. • Task-4: Use DELETE button. It will delete the data which user wants to remove. • Task-5: Use EXIT button. When user want to exit from the GUI created user will simple click on this. • Task-6: Create a database with table named "USER". This table will contain two columns namely "ID" and "NAME". • Task-7: Create the data source using JDBC-ODBC Bridge.	At the end learners will be able to: • Describe and use the JDBC components. • They can use the benefits of JDBC-ODBC Bridge to develop an interactive Java based applications. • Able to demonstrate their understand the meaning of front end and back end.

	• Task-8: Compile and execute your program and verify the connectivity and working of each component given on the GUI.	
4.	Create a java program to perform the following task. • Task-1: Create a layout with three text boxes for Student Name, Roll Number and course information (B.Tech/M.Tech). • Task-2: Create buttons “INSERT”, “UPDATE”, “DELETE” and “EXIT” • Task-3: Insert: to insert information, Delete: to remove information, Update: to update information and Exit: to quite. • Task-4: Create table name student with three field rollnumber, sname and course. • Task-5: Create a data source named studata • Task-6: Create connection between java code and the data source. • Task-7: Compile and execute the java program. • Task-8: Finally, verify your connectivity by using test cases.	At the end learners will be able to: • Describe and use the JDBC components. • They can use the benefits of JDBC-ODBC Bridge to develop an interactive Java based applications. • Able to demonstrate their understand the meaning of front end and back end.
5.	Developing Servlets based applications	At the end learners will be able to: • Describe the working of Servlets. • Explain and demonstrate the architectural aspects involved in Servlets application development. • They will be able to justify the Tasks performed by servlets as a middle layer. • Describe and understand the

		servlet life-cycle.
6.	Developing Servlets and JSP based Application. Using Tomcat to create the database.	At the end learners will be able to: • Describe the working of Servlets and JSP. • Understand the benefits of using dynamic database. • Perform the connectivity of Java applications with Tomcat of any other database.
7.	Developing application using Enterprise Java Beans	At the end learners will be able to: • Describe the working of EJB. • Develop an EJB application. • Explain the benefits of using EJP
8.	Developing Java based application to demonstrate the working of Struts and creating Beans	In process.....
9.	Developing a simple Android based application using SDK.	In process.....
10.	Developing an interactive Android Application using SDK and understanding the working of application developed with the Android Architecture stand points.	In process.....