

**(Established under the Presidency University Act, 2013 of the Karnataka Act 41 of 2013)**

## CSE3146 – Advanced JAVA Programming

## LAB SHEET – 4

**Module 4 – Distributed Programming with servlets and JSP**

**Q1**. To study and implement java servlet to create login and registration servlet necessary for student information management system.

**Solution :**

Create a servlet in eclipse, we need to follow the below steps.

* Create the Dynamic web project.
* Create a servlet
* Add servlet-api.jar Jar file (use build path->add external jar)
* Add mysql-connector Jar file (use build path->add external jar)
* Run the servlet.

1. Create the Dynamic Project: click on File Menu -> New -> Dynamic Web Project and write project name and click next and select Generate web.xml deployment descriptor.

Ex: first

1. Create a servlet : For creating a servlet right click on src in your project -> New-> Servlet and write your servlet name. Ex. Registeruser -> uncheck all the checkboxes except doGet() -> next -> Finish.
2. Add servlet-api.jar file: For adding jar file right click on your project -> Build Path -> Configure Build Path-> Click Add External JARs button -> select servlet-api.jar from Tomcat Server lib folder.
3. Add mysql-connector jar file : For adding jar file right click on your project -> Build Path -> Configure Build Path -> Click Add External JARs Button -> select mysql-connector-j from mysql 8.0 and above connector/j folder.

To connect java application with the mysql database mysql-connector.jar file is required to be loaded. Put this jar file in WEB-INF/lib folder.

1. Run the Servlet: For Run the Servlet start the server and deploy the project. Right Click on your project -> Run As -> Run on Server -> choose tomcat server -> next -> add servlet -> Finish.

How to Configure tomcat server in Eclipse(Onetime Requirement):

For configuring tomcat server in eclipse IDE:

* Click on servers tab at the bottom side of the IDE
* Right click on blank area -> New ->Servers
* Choose tomcat then its version -> next -> click on browse button
* Select the Apache tomcat root folder previous to bin -> next -> Add all -> Finish

Create studentdb database and create table in studentdb database.

**Create Table in student database:**

**CREATE TABLE REGISTERUSER(NAME VARCHAR(100) PRIMARY KEY,**

**PASS VARCHAR(50),** **EMAIL VARCHAR(100), COUNTRY VARCHAR(20));**

**Register.html:**

<html>

<body>

<form action="servlet/Register" method="post">

Name:<input type="text" name="userName"/><br/><br/>

Password:<input type="password" name="userPass"/><br/><br/>

Email Id:<input type="text" name="userEmail"/><br/><br/>

Country:

<select name="userCountry">

<option>India</option>

<option>USA</option>

<option>other</option>

</select>

<br/><br/>

<input type="submit" value="register"/>

</form>

</body>

</html>

**Web.xml:**

<?xml version="1.0" encoding="UTF-8"?>

<servlet>

<servlet-name>Register</servlet-name>

<servlet-class>Register</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Register</servlet-name>

<url-pattern>/servlet/Register</url-pattern>

</servlet-mapping>

<welcome-file-list>

<welcome-file>register.html</welcome-file>

</welcome-file-list>

</web-app>

**Register.java:**

import java.io.\*;

import java.sql.\*;

import javax.servlet.ServletException;

import javax.servlet.http.\*;

public class Register extends HttpServlet {

public void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

String n=request.getParameter("userName");

String p=request.getParameter("userPass");

String e=request.getParameter("userEmail");

String c=request.getParameter("userCountry");

try{

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/studentdb","root","root");

PreparedStatement ps=con.prepareStatement("insert into registeruser values(?,?,?,?)");

ps.setString(1,n);

ps.setString(2,p);

ps.setString(3,e);

ps.setString(4,c);

int i=ps.executeUpdate();

if(i>0)

out.print("You are successfully registered...");

}catch (Exception e2) {System.out.println(e2);}

out.close();

}

}

Q2. To study and implement java servlet to receive data from servlet request header and print the received data over web browser.

1. Create the Dynamic Project: click on File Menu -> New -> Dynamic Web Project and write project name and click next and select Generate web.xml deployment descriptor.

Ex: first

1. Create a servlet : For creating a servlet right click on src in your project -> New-> Servlet and write your servlet name. Ex. Hello -> uncheck all the checkboxes except doGet() -> next -> Finish.
2. Add servlet-api.jar file: For adding jar file right click on your project -> Build Path -> Configure Build Path-> Click Add External JARs button -> select servlet-api.jar from Tomcat Server lib folder.

**ReceiveData.html:**

<form action="servlet/Hello">

Name:<input type="text" name="user"/>

<input type="submit" value="go"/>

</form>

**Web.xml:**

<?xml version="1.0" encoding="UTF-8"?>

<servlet>

<description>This is the description of my J2EE component</description>

<display-name>This is the display name of my J2EE component</display-name>

<servlet-name>Hello</servlet-name>

<servlet-class>Hello</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Hello</servlet-name>

<url-pattern>/servlet/Hello</url-pattern>

</servlet-mapping>

<welcome-file-list>

<welcome-file>ReceiveData.html</welcome-file>

</welcome-file-list>

</web-app>

**Hello.java:**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class Hello extends HttpServlet {

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

String name=request.getParameter("user");

out.print("Hello "+name);

out.close();

}

}

Q3. To study and implement java server page to print factorial value of an integer number, while the input is taken from an HTML form.

1. Create the Dynamic Project: click on File Menu -> New -> Dynamic Web Project and write project name and click next and select Generate web.xml deployment descriptor.

Ex: first

1. Create a servlet : For creating a servlet right click on src in your project -> New-> Servlet and write your servlet name. Ex. Fact -> uncheck all the checkboxes except doGet() -> next -> Finish.
2. Add servlet-api.jar file: For adding jar file right click on your project -> Build Path -> Configure Build Path-> Click Add External JARs button -> select servlet-api.jar from Tomcat Server lib folder.

**Factorial.html:**

<form action="servlet/fact">

Name:<input type="text" name="user"/>

<input type="submit" value="go"/>

</form>

**Web.xml:**

<?xml version="1.0" encoding="UTF-8"?>

<servlet>

<description>This is the description of my J2EE component</description>

<display-name>This is the display name of my J2EE component</display-name>

<servlet-name>Factorial</servlet-name>

<servlet-class>fact</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>fact</servlet-name>

<url-pattern>/servlet/fact</url-pattern>

</servlet-mapping>

<welcome-file-list>

<welcome-file>Factorial.html</welcome-file>

</welcome-file-list>

</web-app>

**fact.java**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class fact extends HttpServlet {

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

int n=Integer.parseInt(request.getParameter("user"));

long f=1;

for(int i=1;i<=n;i++)

f=f\*i;

out.print("Factorial of "+n+"is "+f);

out.close();

}

}

## Record Writing

**Note: Submit the record on or before the due date.**

**Both Soft and Hard copies are required to be submitted.**

# RECORD WRITING INSTRUCTIONS

1. Solve the programming exercise using any IDE (Laptop / Mobile) or using any online compiler.
   1. Students can use online compiler or any preferable platform for the execution.
   2. Mobile users, kindly install JStudio - ide for java https://play.google.com/store/apps/details?id=com.qamar.ide.java&hl=en. This instruction is already given for solving your lab programs. If you haven’t done, please do install, and test the app as soon as possible.
2. While solving your programming exercise, write the code in A4 sheet paper/Record. While writing on the paper, please add these info. “Presidency University” “Department of CSE” “Odd semester 2021- 2022” “MODULE - 3" “Course code : CSE 3146”, Course name : Advanced Java Programming, ID: , NAME: , SEC: , Date: \_
   1. While coding (the soft copy) & writing in the paper/record, all your **CLASS NAME** and the **METHOD NAME** must be appended with your **LAST FOUR DIGIT student ID**. This is mandatory, even while WRITING in the paper/record.

class **sample0161** {

void **Method0161**(parameterlist) {

//method body

}

**For example: If your Registration number is 2021CSE0161 then**

* 1. Take a screenshot of **your program & the output** from your mobile/laptop.
  2. Take a photo of the handwritten program.
  3. Put together (**4. A,B,C**) , **combine as one pdf**, with the file name as your student **registration number(ex. 2021CSE0161.pdf)**, and upload the file in Camu.
  4. The document must be uploaded within the specified time in Camu.

Kindly follow the instructions very carefully so that your submission will be valid.