Diploma Engineering

Laboratory Manual

(Web based Java Programming)
(4350708)

[Computer Engineering, Semester V]

Enrolment No	
Name	
Branch	
Academic Term	
Institute	

Directorate of Technical Education

Gandhinagar - Gujarat

DTE's Vision:

- To provide globally competitive technical education;
- Remove geographical imbalances and inconsistencies;
- Develop student friendly resources with a special focus on girls' education and support to weaker sections;
- Develop programs relevant to industry and create a vibrant pool of technical professionals.

Institute'sVision:

Institute's Mission:

Department's Vision:

Department's Mission:

Name of institute

Certificate

This is to certify that Mr./Ms …….. Enrollment No. …….. of 5th Semester of *Diploma in Computer Engineering* of …….. (GTU Code) has satisfactorily completed the term work in course …….. for the academic year: …… Term: Odd/Even prescribed in the GTU curriculum.

Place:……..

Date: ……..

Signature of Course Faculty **Head of the Department**

Preface

The primary aim of any laboratory/Practical/field work is enhancement of required skills as well as creative ability amongst students to solve real time problems by developing relevant competencies in psychomotor domain. Keeping in view, GTU has designed competency focused outcome-based curriculum -2021 (COGC-2021) for Diploma engineering programmes. In this more time is allotted to practical work than theory. It shows importance of enhancement of skills amongst students and it pays attention to utilize every second of time allotted for practical amongst Students, Instructors and Lecturers to achieve relevant outcomes by performing rather than writing practice in study type. It is essential for effective implementation of competency focused outcome- based Green curriculum-2021. Every practical has been keenly designed to serve as a tool to develop & enhance relevant industry needed competency in each and every student. These psychomotor skills are very difficult to develop through traditional chalk and board content delivery method in the classroom. Accordingly, this lab manual has been designed to focus on the industry defined relevant outcomes, rather than old practice of conducting practical to prove concept and theory.

By using this lab manual, students can read procedure one day in advance to actual performance day of practical experiment which generates interest and also, they can have idea of judgement of magnitude prior to performance. This in turn enhances predetermined outcomes amongst students. Each and every Experiment /Practical in this manual begins by competency, industry relevant skills, course outcomes as well as practical outcomes which serve as a key role for doing the practical. The students will also have a clear idea of safety and necessary precautions to be taken while performing experiment.

This manual also provides guidelines to lecturers to facilitate student-centered lab activities for each practical/experiment by arranging and managing necessary resources in order that the students follow the procedures with required safety and necessary precautions to achieve outcomes. It also gives an idea that how students will be assessed by providing Rubrics.

Web based Java Programming is a set of technologies and frameworks used for developing enterprise-level Java applications. It emphasis on the fundamentals of the client service architecture for web based applications. Java Database Connectivity and Hibernate are technologies used for interacting with databases in Java applications. Servlets and JSPs are technologies used for building dynamic web applications in Java. They are often used together to provide a complete solution for handling web requests and generating dynamic web pages. Web socket programming is a technology used for real-time communication between web clients and servers which provides more efficient and

scalable alternative to traditional HTTP-based communication. Web services are to provide a standardized, platform-independent, and language-independent mechanism for applications to communicate with each other over the internet.

Although we try our level best to design this lab manual, but always there are chances of improvement. We welcome any suggestions for improvement.

Programme Outcomes (POs) to be achieved through practicals of this Course

Following programme outcomes are expected to be achieved through the practical of the course:

1. **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.

2. **Problem analysis**: Identify and analyze well-defined engineering problems using codified standard methods

3. **Design/ development of solutions** Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs

4. **Engineering Tools, Experimentation and Testing:** Apply modern engineering tools and appropriate technique to conduct standard tests and measurements

5. **Engineering practices for society, sustainability and environment:** Apply appropriate technology in context of society, sustainability, environment and ethical practices.

6. **Project Management:** Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities.

7. **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes.

Practical Outcome - Course Outcome matrix

Course Outcomes (COs): Students will be able to

a) Implement basic database operations using JDBC.
b) Develop database-driven Java applications using Hibernate ORM framework.
c) Develop server side programs using Servlets.
d) Develop Java Server Pages application using JSP tags.
e) Develop networked applications in java using using network protocols, socket programming, and related technologies.
f) Develop of simple web service applications using Java technologies

S.No	Practical Outcome	CO1	CO2	CO3	CO4	CO5	CO6
1.	Develop a database application that uses any JDBC driver	√	-	-	-	-	-
2.	Develop a program to present a set of choice for user to select a product and display the price of product.	√	-	-	-	-	-
3.	Develop a simple hibernate Web Application that displays all records stored in a student table having attributes student_id, student_name and student_branch.	-	√	-	-	-	-
4.	Develop a simple hibernate Web Application that displays total number of employees in an organization with its maximum	-	√	-	-	-	-
5.	Write an HTML code to create login form having one submit button, two textboxes labeled as Login name and Password as respectively. Write a Servlet class named as ReadParameter to read these two parameters and display entered parameters values on the page using doGet() or doPost() method when user clicked on submit button.	-	-	√	-	-	-
6.	Create a java application to call one servlet from another servlet.	-	-	√	-	-	-
7.	Create a web form which processes servlet and demonstrates use of cookies and sessions.	-	-	√	-	-	-
8.	Develop a simple JSP program for user registration and then control will be transfer it into second page.	-	-	-	√	-	-
9.	Develop a JSP program to display the grade of a student by accepting the marks of five subjects	-	-	-	√	-	-
10.	Develop a JSP application to insert, update and display record in MySQL database. (Assume suitable database)	-	-	-	√	-	-

11.	Develop a student login application using MVC architecture. Create StudentLogin.jsp, StudentLoginController, StudentLoginDAO and StudentLoginModel and display whether student gets successfully logged in or not.	-	-	-	√	-	-
12.	Write a client server program where client sends two numbers and server responds with square of them.	-	-	-	-	√	-
13.	Develop chat application using socket programming.	-	-	-	-	√	-
14.	Develop a SOAP web service that prints your name using eclipse.	-	-	-	-	-	√

Industry Relevant Skills

The following industry relevant skills of the competency "**Develop java web based applications using Servlet, JSP and Hibernate**" are expected to be developed in the student by undertaking the practical of this laboratory manual.

1. Implement basic database operations using JDBC and Hibernate framework.
2. Develop server side programs using Servlets and JSPs.
3. Develop networked applications in java using using network protocols, socket programming, and related technologies.
4. Develop of simple web service applications using SOAP.

Guidelines to Course Faculty

1. Course faculty should demonstrate experiment with all necessary implementation strategies described in curriculum.
2. Course faculty should explain industrial relevance before starting of each experiment.
3. Course faculty should involve & give opportunity to all students for hands on experience.
4. Course faculty should ensure mentioned skills are developed in the students by asking.
5. Utilise 2 hrs of lab hours effectively and ensure completion of write up with quiz also.
6. Encourage peer to peer learning by doing same experiment through fast learners.

Instructions for Students

1. Organize the work in the group and make record of all necessary outputs.
2. Students shall develop coding standard skill as expected by industries.
3. Student shall attempt to develop related hand-on skills and build confidence.
4. Student shall develop the habits of evolving more ideas, innovations, skills etc.
5. Student shall refer technical magazines and data books.
6. Student should develop habit to submit the practical on date and time.
7. Student should well prepare while submitting write-up of exercise.

A. V. PAREKH TECHNICAL INSTITUTE, RAJKOT

COMPUTER ENGINEERING DEPARTMENT

ASSESSMENT RUBRICS FOR PRACTICAL COMPONENTS

SUBJECT & CODE: WEB BASED JAVA PROGRAMMING (4350708)

- **CONTINUOUS ASSESSMENT (25 Marks):**
  - **Laboratory Work and Questionnaire Component (25 Marks):**

Component	Criteria	Percentage	Assessment
Laboratory Work and Questionnaire	**Excellent**	**91%-100%**	Demonstrates exceptional proficiency in both laboratory work and questionnaire assessments, consistently applying skills and understanding effectively.
	Proficient	**71%-90%**	Shows a strong command of both laboratory work and questionnaire assessments, with minor areas for improvement.
	Satisfactory	**51%-70%**	Achieves a satisfactory level of performance in laboratory work and questionnaire assessments, with room for improvement in some areas.
	Needs Improvement	**31%-50%**	Demonstrates limited proficiency in both laboratory work and questionnaire assessments, with significant areas for improvement.
	Inadequate	**0%-30%**	Fails to meet acceptable standards in both laboratory work and questionnaire assessments; significant improvement is required.

A. V. PAREKH TECHNICAL INSTITUTE, RAJKOT

COMPUTER ENGINEERING DEPARTMENT

ASSESSMENT RUBRICS FOR PRACTICAL COMPONENTS

SUBJECT & CODE: WEB BASED JAVA PROGRAMMING (4350708)

- **END SEMESTER EXAMINATION (25 Marks):**
  - **Viva Examination (25 Marks):**

Component	Criteria	Percentage	Assessment
Viva Examination	**Excellent**	**91%-100%**	Demonstrates exceptional proficiency in the viva exam, displaying an in-depth understanding and providing comprehensive and insightful answers.
	Proficient	**71%-90%**	Displays a strong grasp of the viva exam topics, providing clear and well-reasoned answers, with minor areas for improvement.
	Satisfactory	**51%-70%**	Provides satisfactory responses during the viva exam, covering the essential topics, with room for improvement in some areas.
	Needs Improvement	**31%-50%**	Demonstrates limited understanding of the viva exam topics, providing answers that may lack clarity or depth, with significant areas for improvement.
	Inadequate	**0%-30%**	Fails to meet acceptable standards in the viva exam, providing answers that are unclear, incorrect, or lacking substance; significant improvement is required.

***Note**: This Rubric is applied to the ESE Components of the Courses where End Semester Examination is conducted by Institute Faculty Internally. For the Final Year Courses, ESE Exam is conducted by an External Faculty appointed by university. So, for those courses the marks are converted from the GTU Grade and equally divided into all the COs.*

Continuous Assessment Sheet

Enrolment No: **Name:**

Term:

Sr. No.	Practical Outcome/Title of experiment	Page	Date	Marks (25)	Sign
1	Develop a database application that uses any JDBC driver				
2	Develop a program to present a set of choice for user to select a product and display the price of product.				
3	Develop a simple hibernate Web Application that displays all records stored in a student table having attributes student_id, student_name and student_branch.				
4	Develop a simple hibernate Web Application that displays total number of employees in an organization with its maximum				
5	Write an HTML code to create login form having one submit button, two textboxes labeled as Login name and Password as respectively. Write a Servlet class named as ReadParameter to read these two parameters and display entered parameters values on the page using doGet() or doPost() method when user clicked on submit button.				
6	Create a java application to call one servlet from another servlet.				
7	Create a web form which processes servlet and demonstrates use of cookies and sessions.				
8	Develop a simple JSP program for user registration and then control will be transfer it into second page.				
9	Develop a JSP program to display the grade of a student by accepting the marks of five subjects				
10	Develop a JSP application to insert, update and display record in MySQL database. (Assume suitable database)				
11	Develop a student login application using MVC architecture. Create StudentLogin.jsp, StudentLoginController, StudentLoginDAO and StudentLoginModel and display whether student gets successfully logged in or not.				

12	Write a client server program where client sends two numbers and server responds with square of them.				
13	Develop chat application using socket programming.				
14	Develop a SOAP web service that prints your name using eclipse.				
Total					

Course Outcome Based Practical Evaluation

Title	CO Covered	Marks Attained (A)	Total Marks (B)	CO Weightage for Internal (C)	CO Weightage for Internal (C) Total Marks (A / B) x C
Practical – 1 to 2	**CO1**		**50**	**3**	
Practical – 3 to 4	**CO2**		**50**	**4**	
Practical – 5 to 7	**CO3**		**75**	**5**	
Practical – 8 to 11	**CO4**		**100**	**8**	
Practical – 12 to 13	**CO5**		**50**	**3**	
Practical - 14	**CO6**		**25**	**2**	
				Total	

Date: ……

Practical No: 1.

A. Objective:

Develop a database application that uses any JDBC driver

B. Expected Program Outcomes (POs):

- **PO1:** Basic and Discipline specific knowledge
- **PO2:** Problem analysis
- **PO3:** Design/ development of solutions
- **PO4:** Engineering Tools, Experimentation and Testing
- **PO6:** Project Management
- **PO7:** Life-long learning

C. Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- Setup database and run queries
- Download and install any one JDBC driver.
- Develop and run program using any IDE.

D. Expected Course Outcomes(Cos)

CO1: Implement basic database operations using JDBC

E. Practical Outcome(PRo)

Install any one JDBC driver and configure it. Develop program to connect java with database and run using any IDE.

F. Expected Affective domain Outcome (ADos)

- Follow coding standards.
- Demonstrate working as a leader/ a team member.
- Follow ethical practices.

G. Prerequisite Theory:

JDBC Driver is required to process SQL requests and generate result. The following are the different types of driver available in JDBC.

- Type-1 Driver or JDBC-ODBC bridge
- Type-2 Driver or Native API Partly Java Driver
- Type-3 Driver or Network Protocol Driver
- Type-4 Driver or Thin Driver

Steps to connect a Java Application to Database

The following 5 steps are the basic steps involve in connecting a Java application with Database using JDBC.

1. Register the Driver

2. Create a Connection
3. Create SQL Statement
4. Execute SQL Statement
5. Closing the connection

1. Register the Driver

Class.forName() is used to load the driver class explicitly.

Example:

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

2. Create a Connection

getConnection() method of DriverManager class is used to create a connection

Syntax

```
Connection getConnection(String url)
Connection getConnection(String url, String username, String password)
Connection getConnection(String url, Properties info)
```

Example:

```
Connection con = DriverManager.getConnection
("jdbc:mysql://localhost:3306/dbname ", "username", "password");
```

3. Create SQL Statement

createStatement() method is invoked on current Connection object to create a SQL Statement.

Syntax

```
public Statement createStatement() throws SQLException
```

Example:

```
Statement s=con.createStatement();
```

4. Execute SQL Statement

executeQuery() method of Statement interface is used to execute SQL statements.

Syntax

```
public ResultSet executeQuery(String query) throws SQLException
```

Example

```
ResultSet rs=s.executeQuery("select * from user");
while(rs.next())
{
        System.out.println(rs.getString(1)+" "+rs.getString(2));
}
```

5. Closing the connection

After executing SQL statement you need to close the connection and release the session. The close() method of **Connection** interface is used to close the connection.

Syntax

```
public void close() throws SQLException
```

Example

```
con.close();
```

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable),RAM 2 GB onwards	As per batch size
2	Software: Operating System	Windows/ Linux	
3	Software	JDK 1.8.0 or above, Notepad++ or other Java IDE, Any Database, Any JDBC Driver.	

I. Safety and necessary Precautions followed

- Shutdown computer system properly once the Lab hours is finished.

J. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

K. Input-Output :

L. Practical related Quiz.

i) Write JDBC Driver name and connection URL for following databases

a) Oracle :
b) MYSQL:
c) PostgreSQL:

ii) What is the correct sequence to create a database connection?

i. Import JDBC packages.
ii. Open a connection to the database.
iii. Load and register the JDBC driver.
iv. Execute the statement object and return a query resultset.
v. Create a statement object to perform a query.
vi. Close the resultset and statement objects.
vii. Process the resultset.
viii. Close the connection.

a) i, ii, iii, v, iv, vii, viii, vi
b) i, iii, ii, v, iv, vii, vi, viii
c) ii, i, iii, iv, viii, vii, v, vi
d) i, iii, ii, iv, v, vi, vii, viii

M. References / Suggestions

- https://www.geeksforgeeks.org/establishing-jdbc-connection-in-java/
- https://www.javatpoint.com/steps-to-connect-to-the-database-in-java
- https://www.tutorialspoint.com/jdbc/jdbc-db-connections.htm

Sign with Date

Date: ……

Practical No: 2.

A. Objective:

Develop a program to present a set of choice for user to select a product and display the price of product.

B. Expected Program Outcomes (POs):

- **PO1:** Basic and Discipline specific knowledge
- **PO2:** Problem analysis
- **PO3:** Design/ development of solutions
- **PO4:** Engineering Tools, Experimentation and Testing
- **PO6:** Project Management
- **PO7:** Life-long learning

C. Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- Accept choice from user
- Query to database as per choice of user

D. Expected Course Outcomes (Cos)

CO1: Implement basic database operations using JDBC

E. Practical Outcome (PRo)

Develop a program that connects to a database and fetches the price of the product opted by user.

F. Expected Affective domain Outcome (ADos)

- Follow coding standards.
- Demonstrate working as a leader/ a team member.
- Follow ethical practices.

G. Prerequisite Theory:

Statement objects allow you to execute basic SQL queries and retrieve the results through the ResultSet class. To create a Statement instance, you call the createStatement() method on the Connection object you have retrieved using one of the DriverManager.getConnection() or DataSource.getConnection() methods described earlier.

Once you have a Statement instance, you can execute a SELECT query by calling the executeQuery(String) method with the SQL you want to use. To update data in the database,

use the executeUpdate(String SQL) method. This method returns the number of rows matched by the update statement, not the number of rows that were modified.

If you do not know ahead of time whether the SQL statement will be a SELECT or an UPDATE/INSERT, then you can use the execute(String SQL) method. This method will return true if the SQL query was a SELECT, or false if it was an UPDATE, INSERT, or DELETE statement. If the statement was a SELECT query, you can retrieve the results by calling the getResultSet() method. If the statement was an UPDATE, INSERT, or DELETE statement, you can retrieve the affected rows count by calling getUpdateCount() on the Statement instance.

Here choice can be taken from user via Scanner class or using command line arguement.

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable),RAM 2 GB onwards	As per batch size
2	Software: Operating System	Windows/ Linux	
3	Software	Jdk 1.8.0 or above, Notepad++ or other Java IDE, Any Database, Any JDBC Driver.	

I. Safety and necessary Precautions followed.

- Shutdown computer system properly once the Lab hours is finished.

J. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

K. Input-Output:

L. Practical related Quiz.

i) What is the purpose of following methods of ResultSet?

a) executeQuery() :

b) executeUpdate() :

c) execute() :

d) executeBatch() :

M. References / Suggestions

- https://www.tutorialspoint.com/jdbc/jdbc-select-records.htm

Sign with Date

Date: ……

Practical No: 3.

A. Objective:

Develop a simple hibernate Web Application that displays all records stored in a student table having attributes student_id, student_name and student_branch.

B. Expected Program Outcomes (POs):

- **PO1:** Basic and Discipline specific knowledge
- **PO2:** Problem analysis
- **PO3:** Design/ development of solutions
- **PO4:** Engineering Tools, Experimentation and Testing
- **PO6:** Project Management
- **PO7:** Life-long learning

C. Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- Create table student in database with necessary fields
- Use Hibernate framework to fetch and display all records

D. Expected Course Outcomes (Cos)

CO2: Develop database-driven Java applications using Hibernate ORM framework

E. Practical Outcome (PRo)

Develop a simple hibernate Application that displays all records stored in a student table having attributes student_id, student_name and student_branch.

F. Expected Affective domain Outcome (ADos)

- Follow coding standards.
- Demonstrate working as a leader/ a team member.
- Follow ethical practices.

G. Prerequisite Theory:

Hibernate is a framework which provides some abstraction layer, meaning that the programmer does not have to worry about the implementations, Hibernate does the implementations for you internally like Establishing a connection with the database, writing query to perform CRUD operations etc.

It is a java framework which is used to develop persistence logic. Persistence logic means to store and process the data for long use. More precisely Hibernate is an open-source, non-invasive, light-weight java ORM(Object-relational mapping) framework to develop objects

which are independent of the database software and make independent persistence logic in all JAVA, JEE.

Procedure: Hibernate application for all CRUD Operations

1) Download, install and configure necessary classpath for Hibernate.

Step 1) Write URL https://hibernate.org/orm/releases/ in browser.

Step 2) Click on the "More info" button attached with the latest version.

Step 3) Click on the "Download Zip archive" button on the right side.

Step 4) Hence Hibernate JAR files are downloaded successfully.

Step 5) Extract the downloaded zip file.

Step 6) Click on the extracted file.

Step 7) Click on the lib folder.

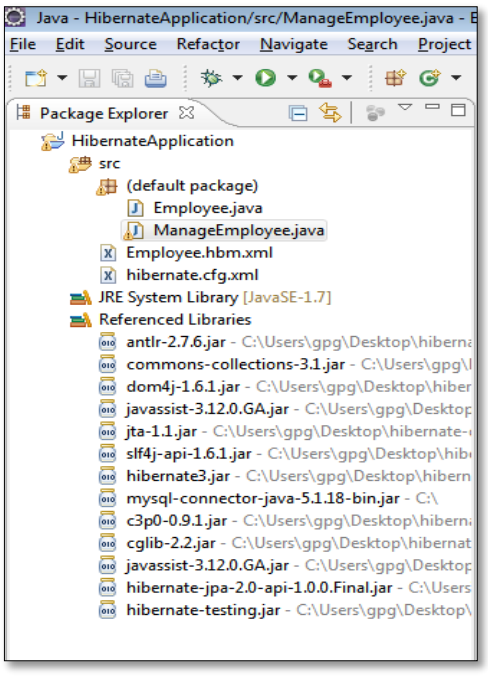
Step 8) Click on all available folder one by one.

Step 9)Right Click on Project->Build path->Configure Build Path->Libraries->add external jars->select all files from particular folder

Step 10) Also add hibernate3.jar to project

Step 11) For database connection , also add mysql-connector-java-5.1.18-bin.jar (download Type 4 driver separately from internet).

All jar files will be added under Reference Libraries folder as shown below



Create remaining files as shown in above hierarchy.

2) **Create POJO Classes**

Employee.java

```
public class Employee {
 private int id;
 private String firstName;
 private String lastName;
 private int salary;

 public Employee() {}
 public Employee(String fname, String lname, int salary) {
   this.firstName = fname;
   this.lastName = lname;
   this.salary = salary;
 }

 public int getId() {
   return id;
 }

 public void setId( int id ) {
   this.id = id;
 }

 public String getFirstName() {
   return firstName;
 }

 public void setFirstName( String first_name ) {
   this.firstName = first_name;
 }

 public String getLastName() {
   return lastName;
 }

 public void setLastName( String last_name ) {
   this.lastName = last_name;
 }

 public int getSalary() {
   return salary;
 }

 public void setSalary( int salary ) {
   this.salary = salary;
 }
```

}3) **Create Database Tables** (here database used in MySql)

```
create table EMPLOYEE (
   id INT NOT NULL auto_increment,
   first_name VARCHAR(20) default NULL,
   last_name  VARCHAR(20) default NULL,
   salary     INT  default NULL,
   PRIMARY KEY (id)
);
```

4) **Create Mapping Configuration File**

The XML configuration file must conform to the Hibernate 3 Configuration DTD, which is available at http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd

To create the configuration file, right click on src - new - file. Now specify the configuration file name e.g. hibernate.cfg.xml.

hibernate.cfg.xml

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE hibernate-configuration SYSTEM "http://www.hibernate.org/dtd/hibernate-
configuration-3.0.dtd">
<hibernate-configuration>
   <session-factory>

      <property name = "hibernate.dialect">
         org.hibernate.dialect.MySQLDialect
      </property>

      <property name = "hibernate.connection.driver_class">
         com.mysql.jdbc.Driver
      </property>

      //employee_info is database name
      <property name = "hibernate.connection.url">
         jdbc:mysql://localhost/employee_info
      </property>

      <property name = "hibernate.connection.username">
         root
      </property>

      <property name = "hibernate.connection.password">

      </property>

      <!-- List of XML mapping files -->
      <mapping resource = "Employee.hbm.xml"/>

   </session-factory>
</hibernate-configuration>
```

You should save the mapping document in a file with the format <classname>.hbm.xml. We saved our mapping document in the file Employee.hbm.xml. Create this file same as configuration file

Employee.hbm.xml

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping>
  <class name = "Employee" table = "EMPLOYEE">

    <meta attribute = "class-description">
      This class contains the employee detail.
    </meta>

    <id name = "id" type = "int" column = "id">
      <generator class="native"/>
    </id>

    <property name = "firstName" column = "first_name" type = "string"/>
    <property name = "lastName" column = "last_name" type = "string"/>
    <property name = "salary" column = "salary" type = "int"/>

  </class>
</hibernate-mapping>
```

5) **Create Application Class**

ManageEmployee.java

```
import java.util.List;

import java.util.Date;

import java.util.Iterator;

import org.hibernate.HibernateException;

import org.hibernate.Session;

import org.hibernate.Transaction;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;
```

```
public class ManageEmployee {
  private static SessionFactory factory;
  public static void main(String[] args) {
    try {
      factory = new Configuration().configure().buildSessionFactory();
    } catch (Throwable ex) {
      System.err.println("Failed to create sessionFactory object." + ex);
     throw new ExceptionInInitializerError(ex);
    }
    ManageEmployee ME = new ManageEmployee();
    /* Add few employee records in database */
    Integer empID1 = ME.addEmployee("Zara", "Ali", 1000);
    Integer empID2 = ME.addEmployee("Daisy", "Das", 5000);
    Integer empID3 = ME.addEmployee("John", "Paul", 10000);

    /* List down all the employees */
    ME.listEmployees();

    /* Update employee's records */
    ME.updateEmployee(empID1, 5000);

    /* Delete an employee from the database */
    ME.deleteEmployee(empID2);

    /* List down new list of the employees */
    ME.listEmployees();
```

```
}
/* Method to CREATE an employee in the database */
public Integer addEmployee(String fname, String lname, int salary){
   Session session = factory.openSession();
   Transaction tx = null;
   Integer employeeID = null;
   try {
      tx = session.beginTransaction();
      Employee employee = new Employee(fname, lname, salary);
      employeeID = (Integer) session.save(employee);
      tx.commit();
   } catch (HibernateException e) {
      if (tx!=null) tx.rollback();
      e.printStackTrace();
   } finally {
      session.close();
   }
   return employeeID;
}

/* Method to  READ all the employees */
public void listEmployees( ){
   Session session = factory.openSession();
   Transaction tx = null;
   try {
      tx = session.beginTransaction();
```

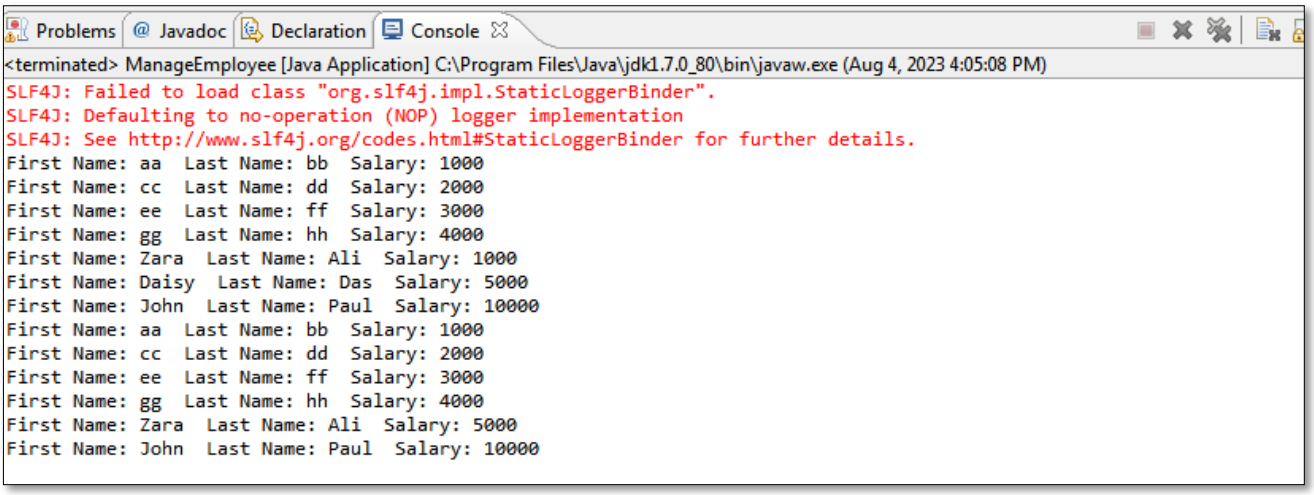
```
      List employees = session.createQuery("FROM Employee").list();
      for (Iterator iterator = employees.iterator(); iterator.hasNext();){
        Employee employee = (Employee) iterator.next();
        System.out.print("First Name: " + employee.getFirstName());
        System.out.print("  Last Name: " + employee.getLastName());
        System.out.println("  Salary: " + employee.getSalary());
      }
      tx.commit();
    } catch (HibernateException e) {
      if (tx!=null) tx.rollback();
      e.printStackTrace();
    } finally {
      session.close();
    }
  }
  /* Method to UPDATE salary for an employee */
  public void updateEmployee(Integer EmployeeID, int salary ){
    Session session = factory.openSession();
    Transaction tx = null;
     try {
      tx = session.beginTransaction();
      Employee employee = (Employee)session.get(Employee.class, EmployeeID);
      employee.setSalary( salary );
            session.update(employee);
      tx.commit();
    } catch (HibernateException e) {
```

```
            if (tx!=null) tx.rollback();
            e.printStackTrace();
         } finally {
            session.close();
         }
      }
      /* Method to DELETE an employee from the records */
      public void deleteEmployee(Integer EmployeeID){
         Session session = factory.openSession();
         Transaction tx = null;
         try {
            tx = session.beginTransaction();
            Employee employee = (Employee)session.get(Employee.class, EmployeeID);
            session.delete(employee);
            tx.commit();
         } catch (HibernateException e) {
            if (tx!=null) tx.rollback();
            e.printStackTrace();
         } finally {
            session.close();
         }
      }
   }
```

6) Compilation and Execution

Right Click on ManageEmployee class-> Run As-> Java Application

Output



H. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable),RAM 2 GB onwards	As per batch size
2	Software: Operating System	Windows/ Linux	
3	Software	Jdk 1.8.0 or above, Notepad++ or other Java IDE, Any Database, Any Database Driver Download the latest version of Hibernate from http://www.hibernate.org/downloads (accumulate all jars in single folder)	

I. Safety and necessary Precautions followed.

- Shutdown computer system properly once the Lab hours is finished.

J. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

K. Input-Output:

L. Practical related Quiz.

i) Which of the following are most common configuration methods of Hibernate Configuration
a) XML Configuration hibernate.cfg.xml
b) Mapping files and XML Configuration hibernate.cfg.xml
c) web.config
d) http.conf

ii) The Hibernate XML configuration file name is?
a) hibernate.cfg.xml
b) hibernate.config.xml
c) hibernate.cg.xml
d) None

M. References / Suggestions

- https://www.tutorialspoint.com/hibernate/hibernate_quick_guide.htm
- https://www.javatpoint.com/example-to-create-hibernate-application-in-eclipse-ide

Sign with Date

Date: ……

Practical No: 4.

A. Objective:

Develop a simple hibernate Web Application that displays total number of employees in an organization with its maximum, minimum, total and average salary of employees.

B. Expected Program Outcomes (POs):

- **PO1:** Basic and Discipline specific knowledge
- **PO2:** Problem analysis
- **PO3:** Design/ development of solutions
- **PO4:** Engineering Tools, Experimentation and Testing
- **PO6:** Project Management
- **PO7:** Life-long learning

C. Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- Create table employee in database with necessary fields
- Use Hibernate framework to fetch and display all records

D. Expected Course Outcomes (Cos)

CO2: Develop database-driven Java applications using Hibernate ORM framework

E. Practical Outcome (PRo)

Develop a simple hibernate Web Application that displays total number of employees in an organization with its maximum, minimum, total and average salary of employees.

F. Expected Affective domain Outcome (ADos)

- Follow coding standards.
- Demonstrate working as a leader/ a team member.
- Follow ethical practices.

G. Prerequisite Theory:

Hibernate is a framework which provides some abstraction layer, meaning that the programmer does not have to worry about the implementations, Hibernate does the implementations for you internally like Establishing a connection with the database, writing query to perform CRUD operations etc.

It is a java framework which is used to develop persistence logic. Persistence logic means to store and process the data for long use. More precisely Hibernate is an open-source, non-invasive, light-weight java ORM(Object-relational mapping) framework to develop objects

which are independent of the database software and make independent persistence logic in all JAVA, JEE.

Procedure:

1) Download, install and configure necessary classpath for Hibernate.

2) Create POJO Classes

3) Create Database Tables

4) Create Mapping Configuration File

5) Create Application Class

6) Compilation and Execution

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable),RAM 2 GB onwards	As per batch size
2	Software: Operating System	Windows/ Linux	
3	Software	Jdk 1.8.0 or above, Notepad++ or other Java IDE, Any Database, Download the latest version of Hibernate from http://www.hibernate.org/downloads	

I. Safety and necessary Precautions followed.

- Shutdown computer system properly once the Lab hours are finished.

J. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

K. Input-Output:

L. Practical related Quiz.

i) In which file database table configuration is stored?
a) .dbm
b) .hbm
c) .ora
d) .sql

M. References / Suggestions

- https://www.tutorialspoint.com/hibernate/hibernate_quick_guide.htm
- https://www.javatpoint.com/example-to-create-hibernate-application-in-eclipse-ide

Sign with Date

Date: ……

Practical No: 5.

A. Objective:

Write an HTML code to create login form having one submit button, two textboxes labeled as Login name and Password as respectively. Write a Servlet class named as ReadParameter to read these two parameters and display entered parameters values on the page using doGet() or doPost() method when user clicked on submit button.

B. Expected Program Outcomes (POs):

- **PO1:** Basic and Discipline specific knowledge
- **PO2:** Problem analysis
- **PO3:** Design/ development of solutions
- **PO4:** Engineering Tools, Experimentation and Testing
- **PO6:** Project Management
- **PO7:** Life-long learning

C. Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- Create HTML form for Login
- Read parameters to servlet using doGet() or doPost() methods

D. Expected Course Outcomes (Cos)

CO3: Develop server side programs using Servlets.

E. Practical Outcome (PRo)

Develop a login form from which read and display parameters in the servlet.

F. Expected Affective domain Outcome (ADos)

- Follow coding standards.
- Demonstrate working as a leader/ a team member.
- Follow ethical practices.

G. Prerequisite Theory:

Parameters may come into our application from the client request, or may be configured through deployment descriptor (web.xml) elements or their corresponding annotations. When you submit a form, form values are sent as request parameters to a web application. In case of a GET request, these parameters are exposed in the URL as name value pairs and in case of POST, parameters are sent within the body of the request.

The API methods to retrieve the request parameters are:

- ServletRequest.getParameterValues(String parmName) o returns a String array with all values present, or null if no value exists for the parameter name.
- ServletRequest.getParameter(String parmName) o returns the first value for the given parameter.
- ServletRequest.getParameterNames() o returns an Enumeration of String objects representing the names of all the parameters in the request. If there are no parameters Enumeration will be empty.
- ServletRequest.getParameterMap() o returns a java.util.Map object, where the keys in the map are of type String (and represent each unique parameter name) and the values in the map of type String array (representing the values for the parameter).

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable),RAM 2 GB onwards	As per batch size
2	Software: Operating System	Windows/ Linux	
3	Software	JDK 7 Eclipse IDE for Java EE Developers Tomcat 7.0	

I. Safety and necessary Precautions followed.

- Shutdown computer system properly once the Lab hours is finished.

J. Procedure to be followed/Source code:

Login.html

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

ReadParameter.java

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class ReadParameter extends HttpServlet
{
        protected void doGet(HttpServletRequest req, HttpServletResponse res) throws
        ServletException,IOException
         {
                res.setContentType("text/html");
                PrintWriter out = res.getWriter();
```

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

```
        }

        protected void doPost(HttpServletRequest req, HttpServletResponse res) throws
        ServletException,IOException
        {
                doGet(req,res);
        }
}
```

Web.xml

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

K. Input-Output:

L. Practical related Quiz.

i) Which of the following method can be used to get the value of form parameter?
a) request.getParameter()
b) request.getParameterValues()
c) request.getParameterNames()
d) None of the above.

ii) Which packages represent interfaces and classes for servlet API?
a) javax.servlet
b) javax.servlet.http
c) Both A & B
d) None of the above

M. References / Suggestions

- https://www.javajee.com/parameters-and-attributes-in-servlet
- https://www.tutorialspoint.com/servlets/servlets-form-data.htm
- https://www.codejava.net/ides/eclipse/how-to-create-deploy-and-run-java-servlet-in-eclipse

Sign with Date

Date: ……

Practical No: 6.

A. Objective:

Create a java application to call one servlet from another servlet.

B. Expected Program Outcomes (POs):

- **PO1:** Basic and Discipline specific knowledge
- **PO2:** Problem analysis
- **PO3:** Design development of solutions
- **PO4:** Engineering Tools, Experimentation and Testing
- **PO6:** Project Management
- **PO7:** Life-long learning

C. Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- Create collaboration between two servlets.

D. Expected Course Outcomes (Cos)

CO3: Develop server side programs using Servlets.

E. Practical Outcome (PRo)

Use RequestDispatcher interface to achieve Servlet Collaboration where one servlet calls another servlet.

F. Expected Affective domain Outcome (ADos)

- Follow coding standards.
- Demonstrate working as a leader/ a team member.
- Follow ethical practices.

G. Prerequisite Theory:

The exchange of information among servlets of a particular Java web application is known as Servlet Collaboration. This enables passing/sharing information from one servlet to the other through method invocations.

What are the principle ways provided by Java to achieve Servlet Collaboration?

The servlet api provides two interfaces namely:

- javax.servlet.RequestDispatcher
- javax.servlet.http.HttpServletResponse

These two interfaces include the methods responsible for achieving the objective of sharing information between servlets.

Using RequestDispatcher Interface

The RequestDispatcher interface provides the option of dispatching the client's request to another web resource, which could be an HTML page, another servlet, JSP etc. It provides the following two methods:

public void forward(ServletRequest request, ServletResponse response)throws ServletException, java.io.IOException:

The forward() method is used to transfer the client request to another resource (HTML file, servlet, jsp etc). When this method is called, the control is transferred to the next resource called. On the other hand, the include() method is used to include the content of the calling file into the called file. After calling this method, the control remains with the calling resource, but the processed output is included into the called resource.

public void include(ServletRequest request, ServletResponse response)throws ServletException, java.io.IOException:

The include() method is used to include the contents of the calling resource into the called one. When this method is called, the control still remains with the calling resource. It simply includes the processed output of the calling resource into the called one.

Code snippet

Considering login application here when user enters username and password

```
if(p.equals("WBJP")
{
        RequestDispatcher rd = request.getRequestDispatcher("servlet2");
        // Getting RequestDispatcher object
        // for collaborating with servlet2 : url pattern of another servlet

        // forwarding the request to servlet2
        rd.forward(request, response);
  }
   else
  {
        out.print("Password mismatch");
        RequestDispatcher rd = request.getRequestDispatcher("/index.html");
        //recalling the same login page (index.html)
          rd.include(request, response);

   }
```

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable),RAM 2 GB onwards	As per batch size
2	Software: Operating System	Windows/ Linux	
3	Software	JDK 7 Eclipse IDE for Java EE Developers Tomcat 7.0	

I. Safety and necessary Precautions followed.

- Shutdown computer system properly once the Lab hours is finished.

J. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

K. Input-Output:

L. Practical related Quiz.

i) What is the purpose of the RequestDispatcher interface in a servlet?
a) To handle HTTP request headers
b) To retrieve client cookies
c) To forward or include requests to other resources
d) To manage servlet configuration settings

ii) Write use of following methods

a) forward() :-

b)include() :-

c) sendRedirect() :-

M. References / Suggestions

- https://www.geeksforgeeks.org/servlet-collaboration-java-using-requestdispatcher-httpservletresponse/
- https://www.geeksforgeeks.org/servlet-forward-and-sendredirect-method-with-example/
- https://www.javatpoint.com/sendRedirect()-method

Sign with Date

Date: ……

Practical No: 7.

A. Objective:

Create a web form which processes servlet and demonstrates use of cookies and sessions.

B. Expected Program Outcomes (POs):

- **PO1:** Basic and Discipline specific knowledge
- **PO2:** Problem analysis
- **PO3:** Design/ development of solutions
- **PO4:** Engineering Tools, Experimentation and Testing
- **PO6:** Project Management
- **PO7:** Life-long learning

C. Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- Session Tracking in servlets

D. Expected Course Outcomes (Cos)

CO3: Develop server side programs using Servlets.

E. Practical Outcome (PRo)

Handling Cookies and a Session in a Java Servlet.

F. Expected Affective domain Outcome (ADos)

- Follow coding standards.
- Demonstrate working as a leader/ a team member.
- Follow ethical practices.

G. Prerequisite Theory:

Session tracking is the process of remembering and documenting customer conversions over time. Session management is another name for it.

The term “stateful web application” refers to a web application that is capable of remembering and recording client conversions over time.

Session Tracking employs Four Different techniques

- Cookies
- Hidden Form Field
- URL Rewriting
- HttpSession

Cookies (javax.servlet.http.Cookie)
Cookies are little pieces of data delivered by the web server in the response header and kept by the browser. Each web client can be assigned a unique session ID by a web server. Cookies are used to keep the session going. Cookies can be turned off by the client.

- How to create Cookie?

```
Cookie ck=new Cookie("user","abc");//creating cookie object
response.addCookie(ck);//adding cookie in the response
```

- How to get Cookies?

```
Cookie ck[]=request.getCookies();
for(int i=0;i<ck.length;i++){
 out.print("<br>"+ck[i].getName()+" "+ck[i].getValue());//printing name and value of cookie
}
```

- How to delete Cookie?

```
Cookie ck=new Cookie("user","");//deleting value of cookie
ck.setMaxAge(0);//changing the maximum age to 0 seconds
response.addCookie(ck);//adding cookie in the response
```

HttpSession

A user session is represented by the HttpSession object. A session is established between an HTTP client and an HTTP server using the HttpSession interface. A user session is a collection of data about a user that spans many HTTP requests.

- How to create Session?

```
HttpSession session = request.getSession( );
Session.setAttribute("username", "password");
```

- Getting a Session

```
HttpSession session = request.getSession();
```

The above code will create a new session in case it doesn't exist and can achieve the same by calling:

```
request.getSession(true)
```

In case we just want to obtain existing session and not create a new one, we need to use:

```
request.getSession(false)
```

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable),RAM 2 GB onwards	As per batch size
2	Software: Operating System	Windows/ Linux	
3	Software	JDK 7 Eclipse IDE for Java EE Developers Tomcat 7.0	

I. Safety and necessary Precautions followed.

- Shutdown computer system properly once the Lab hours is finished.

J. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

K. Input-Output:

L. Practical related Quiz.

i) Which method in session tracking is used in a bit of information that is sent by a web server to a browser and which can later be read back from that browser?
a) HttpSession
b) URL rewriting
c) Cookies
d) Hidden form fields

ii) Which of the following code is used to get an attribute in a HTTP Session object in servlets?
a) session.getAttribute(String name)
b) session.alterAttribute(String name)
c) session.updateAttribute(String name)
d) session.setAttribute(String name)

M. References / Suggestions

- https://www.geeksforgeeks.org/servlet-session-tracking/
- https://www.geeksforgeeks.org/servlet-cookies/
- https://www.javatpoint.com/cookies-in-servlet
- https://www.javatpoint.com/http-session-in-session-tracking
- https://www.baeldung.com/java-servlet-cookies-session

Sign with Date

Date: ……

Practical No: 8.

A. Objective:

Develop a simple JSP program for user registration and then control will be transfer it into second page.

B. Expected Program Outcomes (POs):

- **PO1:** Basic and Discipline specific knowledge
- **PO2:** Problem analysis
- **PO3:** Design/ development of solutions
- **PO4:** Engineering Tools, Experimentation and Testing
- **PO6:** Project Management
- **PO7:** Life-long learning

C. Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- Develop a simple JSP program using scripting elements

D. Expected Course Outcomes (Cos)

CO4: Develop Java Server Pages application using JSP tags

E. Practical Outcome (PRo)

- Develop a simple JSP program for user registration and transfer control to another page.

F. Expected Affective domain Outcome (ADos)

- Follow coding standards.
- Demonstrate working as a leader/ a team member.
- Follow ethical practices.

G. Prerequisite Theory:

JSP technology is used to create dynamic web applications. JSP pages are easier to maintain than a Servlet. A JSP page contains tags. The tags are in the form of scripting tags, directive tags and comments.

Types of Scripting Tags
There are three type of scripting tags:
1. Scriptlet tag
2. Declaration tag
3. Expression tag

JSP Scriptlet Tag

Scriptlet Tag allows you to write java code inside JSP page. Scriptlet tag implements the _jspService method functionality by writing script/java code.

Syntax of Scriptlet Tag is as follows :
<% JAVA CODE %>

JSP Declaration Tag

We know that at the end a JSP page is translated into Servlet class. So when we declare a variable or method in JSP inside Declaration Tag, it means the declaration is made inside the Servlet class but outside the service(or any other) method. You can declare static member, instance variable and methods inside Declaration Tag.

Syntax of Declaration Tag :
<%! declaration %>

JSP Expression Tag

Expression Tag is used to print out java language expression that is put between the tags. An expression tag can hold any java language expression that can be used as an argument to the out.print() method.

Syntax of Expression Tag
<%= Java Expression %>

When the Container sees this
<%= (2*5) %>
It turns it into this:
out.print((2*5));
Note: Never end an expression with semicolon inside Expression Tag. Like this:
<%= (2*5); %>

Code Snippet

index.html

```
<html>
 <body>
       <form action="second.jsp" method="post">
         Enter your Name: <input type="text" name="name">
         <!—more registration fields -->
         <input type="submit">
      </form>
   </body>
</html>
```

Second.jsp

```
<html>
   <body>
      <%
        String name=request.getParameter("name");

         out.println("You Entered Name : "+name+"<br>");
      %>
   </body>
</html>
```

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable),RAM 2 GB onwards	As per batch size
2	Software: Operating System	Windows/ Linux	
3	Software	JDK 7 Eclipse IDE for Java EE Developers Tomcat 7.0	

I. Safety and necessary Precautions followed.

- Shutdown computer system properly once the Lab hours is finished.

J. Procedure to be followed/Source code:

.....

.....

.....

.....

.....

.....

.....

.....

.....

K. Input-Output:

L. Practical related Quiz.

i) Which of the scripting of JSP not putting content into service method of the converted servlet?
a) Declarations
b) Scriptlets
c) Expressions
d) None of the above

ii) An _____ tag can hold any Java language expression that can be used as an argument to out.println() method.?
a) Expression tag
b) Declaration Tag
c) Directive Tag
d) Scriplet Tag

M. References / Suggestions

- https://www.geeksforgeeks.org/introduction-to-jsp/
- https://ecomputernotes.com/jsp/jsp-elements/jsp-scripting-elements
- https://www.studytonight.com/jsp/jsp-scripting-element.php
- https://www.javatpoint.com/jsp-scriptlet-tag

Sign with Date

Date: ……

Practical No: 9.

A. Objective:

Develop a JSP program to display the grade of a student by accepting the marks of five subjects.

B. Expected Program Outcomes (POs):

- **PO1:** Basic and Discipline specific knowledge
- **PO2:** Problem analysis
- **PO3:** Design/ development of solutions
- **PO4:** Engineering Tools, Experimentation and Testing
- **PO6:** Project Management
- **PO7:** Life-long learning

C. Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- Develop a simple JSP program using scripting elements.

D. Expected Course Outcomes (Cos)

CO4: Develop Java Server Pages application using JSP tags

E. Practical Outcome (PRo)

- Develop a simple JSP program to accept marks from user and display its grade accordingly.

F. Expected Affective domain Outcome (ADos)

- Follow coding standards.
- Demonstrate working as a leader/ a team member.
- Follow ethical practices.

G. Prerequisite Theory:

JSP technology is used to create dynamic web applications. JSP pages are easier to maintain than a Servlet. A JSP page contains tags. The tags are in the form of scripting tags, directive tags and comments.

Types of Scripting Tags
There are three type of scripting tags:
1. Scriptlet tag
2. Declaration tag
3. Expression tag

JSP Scriptlet Tag
Scriptlet Tag allows you to write java code inside JSP page. Scriptlet tag implements the _jspService method functionality by writing script/java code.

Syntax of Scriptlet Tag is as follows :
<% JAVA CODE %>

JSP Declaration Tag
We know that at the end a JSP page is translated into Servlet class. So when we declare a variable or method in JSP inside Declaration Tag, it means the declaration is made inside the Servlet class but outside the service(or any other) method. You can declare static member, instance variable and methods inside Declaration Tag.

Syntax of Declaration Tag :
<%! declaration %>

JSP Expression Tag
Expression Tag is used to print out java language expression that is put between the tags. An expression tag can hold any java language expression that can be used as an argument to the out.print() method.

Syntax of Expression Tag
<%= Java Expression %>

When the Container sees this
<%= (2*5) %>
It turns it into this:
out.print((2*5));
Note: Never end an expression with semicolon inside Expression Tag. Like this:
<%= (2*5); %>

Code Snippet

Marks.html
```
<form action="marks.jsp" method="get">
        Enter Makrs in subject1 : <input type="text" name="sub1"> <br><br>
        <!—add five such subjects -->
        <input type="submit">
 </form>
```

Marks.jsp
```
<html>
<body>
      <%
          int s1=Integer.parseInt(request.getParameter("sub1"));
         //code to calculate grade of the student
       %>
</body>
</html>
```

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable),RAM 2 GB onwards	As per batch size
2	Software: Operating System	Windows/ Linux	
3	Software	JDK 7 Eclipse IDE for Java EE Developers Tomcat 7.0	

I. Safety and necessary Precautions followed.

- Shutdown computer system properly once the Lab hours is finished.

J. Procedure to be followed/Source code:

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

K. Input-Output:

L. Practical related Quiz.

i) Which implicit object of JSP is associated with the Output Stream of response object?
a) response
b) write
c) responseWriter
d) out

ii) Given a request with two parameters: one named "first" represents a user's first name and another named "last" represents his last name. Which JSP scriplet code outputs these parameter values?
a) <% out.println(request.getParameter("first")); out.println(request.getParameter("last")); %>
b) <% out.println(apps.getInitParameter("first")); out.println(apps.getInitParameter("last")); %>
c) <% println(request.getParameter("first")); println(request.getParameter("last")); %>
d) <% println(application.getInitParameter("first")); println(application.getInitParameter("last")); %>

M. References / Suggestions

- https://www.geeksforgeeks.org/introduction-to-jsp/
- https://ecomputernotes.com/jsp/jsp-elements/jsp-scripting-elements
- https://www.studytonight.com/jsp/jsp-scripting-element.php
- https://www.javatpoint.com/jsp-scriptlet-tag

Sign with Date

Date: ……

Practical No: 10.

A. Objective:

Develop a JSP application to insert, update and display record in MySQL database. (Assume suitable database)

B. Expected Program Outcomes (POs):

- **PO1:** Basic and Discipline specific knowledge
- **PO2:** Problem analysis
- **PO3:** Design/ development of solutions
- **PO4:** Engineering Tools, Experimentation and Testing
- **PO6:** Project Management
- **PO7:** Life-long learning

C. Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- Access database using JSP.

D. Expected Course Outcomes (Cos)

CO4: Develop Java Server Pages application using JSP tags

E. Practical Outcome (PRo)

- Develop a simple JSP application to perform CRUD operations on database.

F. Expected Affective domain Outcome (ADos)

- Follow coding standards.
- Demonstrate working as a leader/ a team member.
- Follow ethical practices.

G. Prerequisite Theory:

JSP technology is used to create dynamic web applications. JSP pages are easier to maintain than a Servlet. A JSP page contains tags. The tags are in the form of scripting tags, directive tags and comments.

Types of Scripting Tags
There are three type of scripting tags:
1. Scriptlet tag
2. Declaration tag
3. Expression tag

JSP Scriptlet Tag
Scriptlet Tag allows you to write java code inside JSP page. Scriptlet tag implements the _jspService method functionality by writing script/java code.

Syntax of Scriptlet Tag is as follows :
<% JAVA CODE %>

JSP Declaration Tag
We know that at the end a JSP page is translated into Servlet class. So when we declare a variable or method in JSP inside Declaration Tag, it means the declaration is made inside the Servlet class but outside the service(or any other) method. You can declare static member, instance variable and methods inside Declaration Tag.

Syntax of Declaration Tag :
<%! declaration %>

JSP Expression Tag
Expression Tag is used to print out java language expression that is put between the tags. An expression tag can hold any java language expression that can be used as an argument to the out.print() method.

Syntax of Expression Tag
<%= Java Expression %>

When the Container sees this
<%= (2*5) %>
It turns it into this:
out.print((2*5));
Note: Never end an expression with semicolon inside Expression Tag. Like this:
<%= (2*5); %>

Code Snippet (to insert into a database)

index.html

```
<html>
<body>
<form method="post" action="process.jsp">
First name:<br>
<input type="text" name="first_name">
<br>
Last name:<br>
<input type="text" name="last_name">
<br>
City name:<br>
<input type="text" name="city_name">
<br>
Email Id:<br>
```

```
<input type="email" name="email">
<br><br>
<input type="submit" value="submit">
</form>
</body>
</html>
```

process.jsp

```
<%@page import="java.sql.*,java.util.*"%>
<%
String first_name=request.getParameter("first_name");
String last_name=request.getParameter("last_name");
String city_name=request.getParameter("city_name");
String email=request.getParameter("email");
try
{
        Class.forName("com.mysql.jdbc.Driver");
          Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/test",
"root", "");
          Statement st=conn.createStatement();
          int i=st.executeUpdate("insert into
users(first_name,last_name,city_name,email)values('"+first_name+"','"+last_name+"','"+city_n
ame+"','"+email+"')");
       out.println("Data is successfully inserted!");
       }
       catch(Exception e)
       {
       System.out.print(e);
       e.printStackTrace();
       }
 %>
```

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable),RAM 2 GB onwards	As per batch size
2	Software: Operating System	Windows/ Linux	
3	Software	JDK 7 Eclipse IDE for Java EE Developers Tomcat 7.0 Any database (MySQL) with appropriate driver	

I. Safety and necessary Precautions followed.

- Shutdown computer system properly once the Lab hours are finished.

J. Procedure to be followed/Source code:

K. Input-Output:

L. Practical related Quiz.

i) Which is an example of the syntax used to import a class to JSP?

a) <% page import="java.sql.*" %>
b) <%@ page import="java.sql.*" @%>
c) <% import="java.sql.*" %>
d) <%@ page import="java.sql.*" %>

M. References / Suggestions

- https://www.geeksforgeeks.org/introduction-to-jsp/
- https://ecomputernotes.com/jsp/jsp-elements/jsp-scripting-elements
- https://www.studytonight.com/jsp/jsp-scripting-element.php
- https://www.javatpoint.com/jsp-scriptlet-tag

Sign with Date

Date: ……

Practical No: 11.

A. Objective:

Develop a student login application using MVC architecture. Create StudentLogin.jsp, StudentLoginController, StudentLoginDAO and StudentLoginModel and display whether student gets successfully logged in or not.

B. Expected Program Outcomes (POs):

- **PO1:** Basic and Discipline specific knowledge
- **PO2:** Problem analysis
- **PO3:** Design/ development of solutions
- **PO4:** Engineering Tools, Experimentation and Testing
- **PO6:** Project Management
- **PO7:** Life-long learning.

C. Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- Apply MVC to Web programming using java.

D. Expected Course Outcomes (Cos)

CO4: Develop Java Server Pages application using JSP tags

E. Practical Outcome (PRo)

- Develop a web application following MVC architecture having controller, DAO and Model

F. Expected Affective domain Outcome (ADos)

- Follow coding standards.
- Demonstrate working as a leader/ a team member.
- Follow ethical practices.

G. Prerequisite Theory:

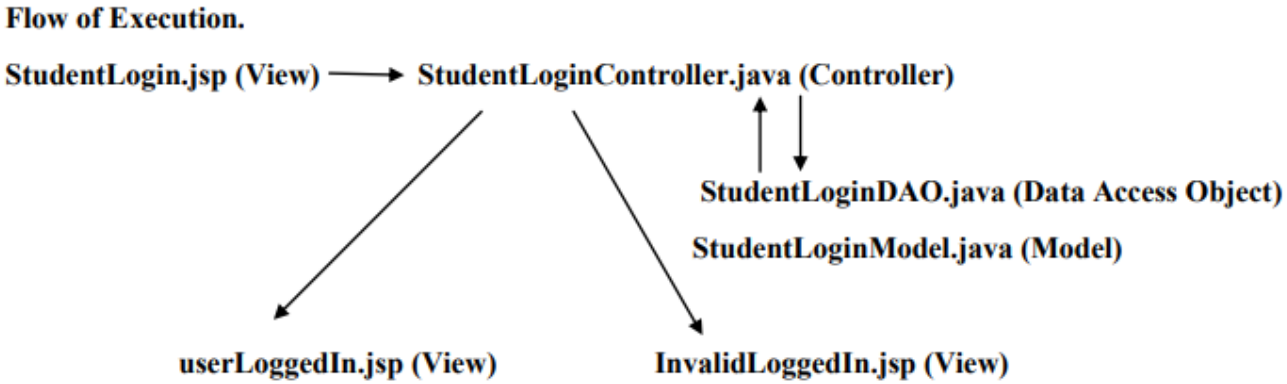
The Model View Controller (MVC) design pattern specifies that an application consist of a data model, presentation information, and control information.

The Model is an object to carry application data.

The View represents the output of the application or the user interface. It displays the data fetched from the model layer by the controller and presents the data to the user whenever asked for. It receives all the information it needs from the controller and it doesn't need to interact with the business layer directly.

The Controller is like an interface between Model and View. It receives the user requests from the view layer and processes them, including the necessary validations. The requests are then sent to the model for data processing. Once they are processed, the data is again sent back to the controller and then displayed on the view.

DAO stands for Data Access Object.DAO is a class that usually has the CRUD operations like save, update, delete. DAO is just an object that holds data.



H. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable),RAM 2 GB onwards	As per batch size
2	Software: Operating System	Windows/ Linux	
3	Software	JDK 7 Eclipse IDE for Java EE Developers Tomcat 7.0 Any database (MySQL) Respective database driver	

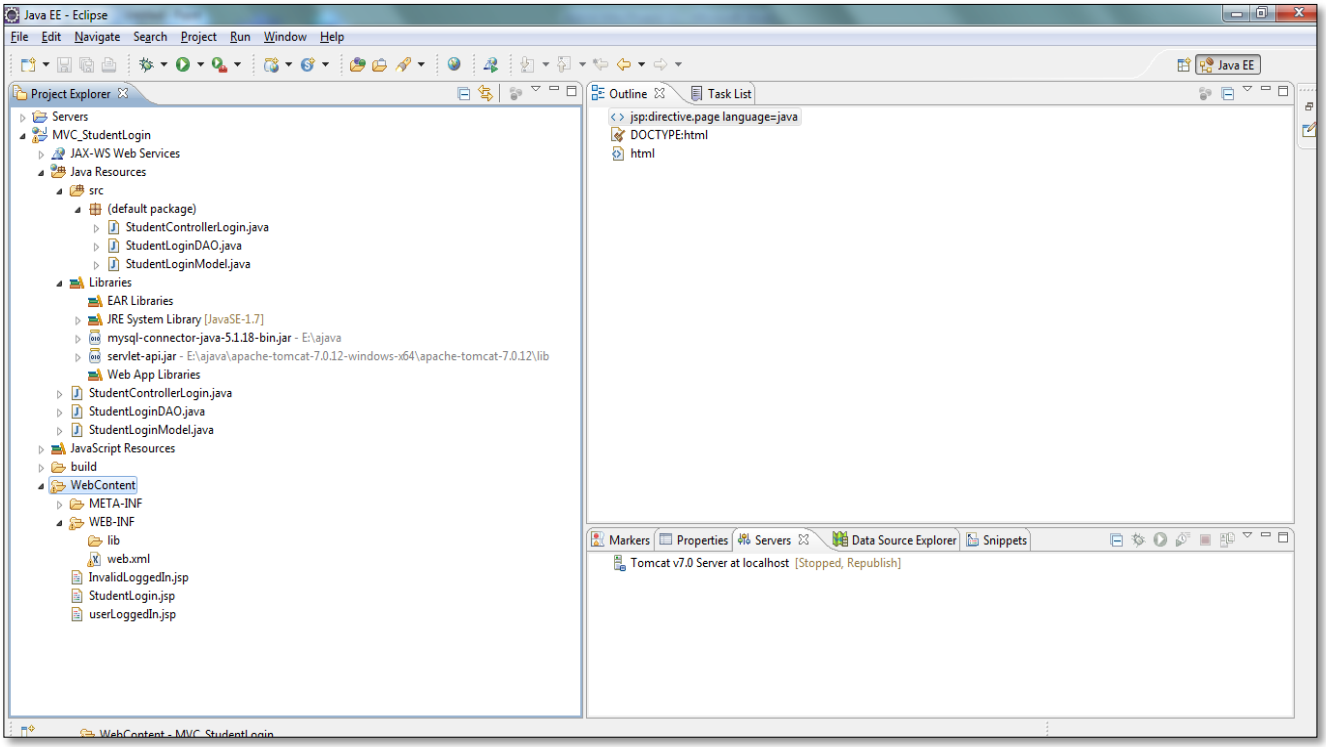
I. Safety and necessary Precautions followed.

- Shutdown computer system properly once the Lab hours is finished.

J. Procedure to be followed/Source code:

Database prerequisites: Table named login having fields username and password

Create files as following



StudentLogin.jsp (View)

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<html>
    <body>
    <form action="doLogin">
         Please enter your username
        <input type="text" name="unm"/><br>

         Please enter your password
         <input type="password" name="pswd"/>

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

      </form>
      </body>
  </html>
```

StudentLoginController.java (Controller)

```
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class StudentLoginController extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException
{
		doPost(request,response);
}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException
{
		//fetching parameters from StudentLogin.jsp
		String username = request.getParameter("unm");
		 String password = request.getParameter("pswd");

		//setting parameters to a POJO (model)
		  StudentLoginModel m = new StudentLoginModel();
		  m.setUsername(username);
		  m.setPassword(password);

		  //calling Data Access Object Layer to check the credentials
		  StudentLoginDAO d = new StudentLoginDAO();
		  m = d.checkLogin(m);

		if(m.isValid())
		{
		     //creating a session
		     HttpSession session = request.getSession();
		     session.setAttribute("username", username);

		    //redirecting page to userLoggedIn.jsp
		     response.sendRedirect("userLoggedIn.jsp");
		}
		else
		{
			response.sendRedirect("InvalidLoggedIn.jsp");
		}
```

```
    }
}
```

StudentLoginDAO.java (Data Access Object)

```
import java.sql.Connection;
 import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class StudentLoginDAO
{
    Connection con=null;
    Statement stmt=null;
    ResultSet rs=null;
    int count;

    public StudentLoginModel checkLogin(StudentLoginModel m)
    {
        String username = m.getUsername();
        String password = m.getPassword();
        System.out.println(username+" "+password);
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            con=DriverManager.getConnection("jdbc:mysql://localhost:3306/login","root","");
            System.out.println("connection successfull");
            stmt = con.createStatement(); System.out.println("Reading from table...");

            String sql = "select count(*) from login_details where username = '"+username+"' and
            password = '"+password+"'";
            rs = stmt.executeQuery(sql); while(rs.next())
            {
                count = rs.getInt(1); System.out.println(count);
            }
            if(count == 1)
            {
                m.setValid(true);
            }
            else
            {
                m.setValid(false);
            }
}
 catch(ClassNotFoundException e)
```

```
{
                System.out.println(e);
}
catch(SQLException e)
{
                System.out.println(e);
}
return m;
}
}
```

StudentLoginModel.java (Model)

```
public class StudentLoginModel
{
 private String username;
private String password;
public boolean valid;
//getters and setters
public String getUsername()
{
                return username;
}
public void setUsername(String username)
{
                 this.username = username;
}
public String getPassword()
{
                 return password;
}
public void setPassword(String password)
{
                this.password = password;
}
public boolean isValid()
{
                 return valid;
}
public void setValid(boolean valid)
{
                this.valid = valid;
}
}
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
<servlet>
<servlet-name>S1</servlet-name>
<servlet-class>StudentLoginController</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>S1</servlet-name>
<url-pattern>/doLogin</url-pattern>
</servlet-mapping>
</web-app>
```

userLoggedIn.jsp (View)

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<% String user = (String ) session.getAttribute("username"); %>

Welcome <%= user %>
</body>
</html>
```

InvalidLoggedIn.jsp (View)

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h3>Sorry.. You are not registered...!!</h3>
</body>
</html>
```

K. Input-Output:

L. Practical related Quiz.

i) Define MVC

M. References / Suggestions

- https://krazytech.com/programs/a-login-application-in-java-using-model-view-controllermvc-design-pattern

Sign with Date

Date: ……

Practical No: 12.

A. Objective:

Write a client server program where client sends two numbers and server responds with square of them.

B. Expected Program Outcomes (POs):

- **PO1:** Basic and Discipline specific knowledge
- **PO2:** Problem analysis
- **PO3:** Design/ development of solutions
- **PO4:** Engineering Tools, Experimentation and Testing
- **PO6:** Project Management
- **PO7:** Life-long learning

C. Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- Develop one way client server program using socket programming.

D. Expected Course Outcomes (Cos)

CO5: Develop networked applications in java using using network protocols, socket programming, and related technologies.

E. Practical Outcome (PRo)

Develop a program where client sends a request and server responds with result.

F. Expected Affective domain Outcome (ADos)

- Follow coding standards.
- Demonstrate working as a leader/ a team member.
- Follow ethical practices.

G. Prerequisite Theory:

Java Socket programming is used for communication between the applications running on different JRE. Socket and ServerSocket classes are used for connection-oriented socket programming and DatagramSocket and DatagramPacket classes are used for connection-less socket programming. The client in socket programming must know two information: IP Address of Server, and Port number.

Here, we are going to make one-way client and server communication. In this application, client sends a message to the server, server reads the message and prints it. Here, two classes are being used: Socket and ServerSocket. The Socket class is used to communicate client and server. Through this class, we can read and write message. The ServerSocket class is used at server-side. The accept()

method of ServerSocket class blocks the console until the client is connected. After the successful connection of client, it returns the instance of Socket at server-side.

Creating Server:

To create the server application, we need to create the instance of ServerSocket class. Here, we are using 6666 port number for the communication between the client and server. You may also choose any other port number. The accept() method waits for the client. If clients connects with the given port number, it returns an instance of Socket.

ServerSocket ss=new ServerSocket(6666);

Socket s=ss.accept();//establishes connection and waits for the client

Creating Client:

To create the client application, we need to create the instance of Socket class. Here, we need to pass the IP address or hostname of the Server and a port number. Here, we are using "localhost" because our server is running on same system.

Socket s=new Socket("localhost",6666);

Client Side Implementation

```
// Java program to illustrate Client Side Programming
// for Simple Calculator using TCP
import java.io.*;
import java.net.*;
import java.util.*;

public class Calc_Client
{
	public static void main(String[] args) throws IOException
	{
		InetAddress ip = InetAddress.getLocalHost();
		int port = 4444;
		Scanner sc = new Scanner(System.in);

		// Step 1: Open the socket connection.
		Socket s = new Socket(ip, port);

		// Step 2: Communication-get the input and output stream
```

```
            DataInputStream dis = new DataInputStream(s.getInputStream());
            DataOutputStream dos = new DataOutputStream(s.getOutputStream());

            while (true)
            {
                // Enter the equation in the form-
                // "operand1 operation operand2"
                System.out.print("Enter the equation in the form: ");
                System.out.println("'operand operator operand'");

                String inp = sc.nextLine();

                if (inp.equals("bye"))
                    break;

                // send the equation to server
                dos.writeUTF(inp);

                // wait till request is processed and sent back to client
                String ans = dis.readUTF();
                System.out.println("Answer=" + ans);
            }
        }
}
```

Server-Side Programming

```
// Java program to illustrate Server Side Programming
// for Simple Calculator using TCP
import java.io.*;
import java.net.*;
import java.util.*;
```

```
public class Calc_Server
{
	public static void main(String args[]) throws IOException
	{

		// Step 1: Establish the socket connection.
		ServerSocket ss = new ServerSocket(4444);
		Socket s = ss.accept();

		// Step 2: Processing the request.
		DataInputStream dis = new DataInputStream(s.getInputStream());
		DataOutputStream dos = new DataOutputStream(s.getOutputStream());

		while (true)
		{
			// wait for input
			String input = dis.readUTF();

			if(input.equals("bye"))
				break;

			System.out.println("Equation received:-" + input);
			int result;

			// Use StringTokenizer to break the equation into operand and
			// operation
			StringTokenizer st = new StringTokenizer(input);

			int oprnd1 = Integer.parseInt(st.nextToken());
```

```
                String operation = st.nextToken();
                int oprnd2 = Integer.parseInt(st.nextToken());

                // perform the required operation.
                if (operation.equals("+"))
                {
                        result = oprnd1 + oprnd2;
                }
                else if (operation.equals("-"))
                {
                        result = oprnd1 - oprnd2;
                }
                else if (operation.equals("*"))
                {
                        result = oprnd1 * oprnd2;
                }
                else
                {
                        result = oprnd1 / oprnd2;
                }
                System.out.println("Sending the result...");

                // send the result back to the client.
                dos.writeUTF(Integer.toString(result));
            }
        }
}
```

Note: In order to test the above programs on the system, please make sure that you run the server program first and then the client one. Make sure you are in the client console and from there enter the equation in the format-“operand1 operator operand2” and press Enter. Answer to the requested equation will be shown in the client console only. Finally to terminate the communication, type “bye” (without quotes) and hit enter.

Client side output:

Enter the equation in the form: 'operand operator operand'

5 * 6

Answer=30

Enter the equation in the form: 'operand operator operand'

5 + 6

Answer=11

Enter the equation in the form: 'operand operator operand'

9 / 3

Answer=3

Server side output:

Equation received:-5 * 6

Sending the result...

Equation received:-5 + 6

Sending the result...

Equation received:-9 / 3

Sending the result...

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable),RAM 2 GB onwards	As per batch size
2	Software: Operating System	Windows/ Linux	
3	Software	Jdk 1.8.0 or above, Notepad++ or other Java IDE	

I. Safety and necessary Precautions followed.

- Shutdown computer system properly once the Lab hours is finished.

J. Procedure to be followed/Source code:

K. Input-Output:

L. Practical related Quiz.

a. A port number in java is defined as _____ integer?

a) 16 bit b) 24 bit c) 48 bit d) none of the above.

b. Network programming in any language definitely needs to deal with _____ and _____

a) user names and port numbers
b) ip address and link-layer address
c)IP addresses and port numbers
d) none of the above

M. References / Suggestions

- https://www.javatpoint.com/socket-programming
- https://www.codejava.net/java-se/networking/java-socket-client-examples-tcp-ip
- https://www.geeksforgeeks.org/socket-programming-in-java/
- https://www.javatpoint.com/socket-programming
- https://examples.javacodegeeks.com/java-socket-programming/
- https://www.geeksforgeeks.org/establishing-the-two-way-communication-between-server-and-client-in-java/
- https://www.geeksforgeeks.org/simple-calculator-using-tcp-java/

Sign with Date

Date: ……

Practical No: 13.

A. Objective:

Develop chat application using socket programming

B. Expected Program Outcomes (POs):

- **PO1:** Basic and Discipline specific knowledge
- **PO2:** Problem analysis
- **PO3:** Design/ development of solutions
- **PO4:** Engineering Tools, Experimentation and Testing
- **PO6:** Project Management
- **PO7:** Life-long learning

C. Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- Develop two way client server program using socket programming.

D. Expected Course Outcomes (Cos)

CO5: Develop networked applications in java using using network protocols, socket programming, and related technologies.

E. Practical Outcome (PRo)

Develop a program where client and server can communicate simultaneously.

F. Expected Affective domain Outcome (ADos)

- Follow coding standards.
- Demonstrate working as a leader/ a team member.
- Follow ethical practices.

G. Prerequisite Theory:

It is possible to send data from the server and receive a response from the client. Similarly, the client can also send and receive data to-and-from.

Below are the various steps to do so:

- We need additional streams both at server and client. For example, to receive data into the server, it is a better idea to use a `BufferedReader` object, as shown in the following code snippet:
  InputStream obj = s.getInputStream();

BufferedReader br = new BufferedReader(new InputStreamReader(obj);

- Then read() or readLine() methods of the BufferedReader object can be used to read data. To send data from the client we can take the help of the DataOutputStream class as shown in the following code snippet:
  OutputStream obj + s.getOutputStrean();

  DataOutputStream dos = new DataOutputStream(obj);

- Then, the writeBytes() method of the DataOutputStream class can be used to send strings in the form of a group of bytes. To establish the two-way communication between a client and server perform the following steps:

  Creating the Server Program: Create a server class which receives data from the client using a BufferedReader object and then sends a reply to the client using a PrintStream object.

  Creating the Client Program: Create a client which first connects to a server, then starts the communication by sending a string to the server. The server sends a response to the client. When 'exit' is typed at the client side, the program terminates.

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable),RAM 2 GB onwards	As per batch size
2	Software: Operating System	Windows/ Linux	
3	Software	Jdk 1.8.0 or above, Notepad++ or other Java IDE	

I. Safety and necessary Precautions followed.

- Shutdown computer system properly once the Lab hours is finished.

J. Procedure to be followed/Source code:

……

……

……

……

……

K. Input-Output:

L. Practical related Quiz.

a) What is the output of the program ?

```
class Message {
    public static void main ( String args [ ] )  {
            String servername = "servername";
            int port = portnumber;
            Socket client = new Socket(servername,portnumber);
            OutputStream out = client.getOutputStream();
            DataOutputStream stream = new DataOutputStream(out);
            stream.write("Message for client");
            InputStream in = client.getInputStream();
            DataInputStream input = new DataInputStream(in);
            System.out.println(in.ReadUTF()):
            client.close();

    } }
```

a) It sends a message to client
b) Receives the message from client
c) Compilation error.
d) Sends and receives messages for clients.

M. References / Suggestions

- https://www.javatpoint.com/socket-programming
- https://www.codejava.net/java-se/networking/java-socket-client-examples-tcp-ip
- https://www.geeksforgeeks.org/socket-programming-in-java/
- https://www.javatpoint.com/socket-programming
- https://examples.javacodegeeks.com/java-socket-programming/
- https://www.geeksforgeeks.org/establishing-the-two-way-communication-between-server-and-client-in-java/

Sign with Date

Date: ……

Practical No: 14.

A. Objective:

Develop a SOAP web service that prints your name using eclipse.

B. Expected Program Outcomes (POs):

- **PO1:** Basic and Discipline specific knowledge
- **PO2:** Problem analysis
- **PO3:** Design/ development of solutions
- **PO4:** Engineering Tools, Experimentation and Testing
- **PO6:** Project Management
- **PO7:** Life-long learning

C. Expected Skills to be developed based on competency:

The practical is expected to develop the following skills:

- Develop simple web service applications using Simple Object Access Protocol.

D. Expected Course Outcomes (Cos)

CO6: Develop of simple web service applications using Java technologies.

E. Practical Outcome (PRo)

Develop a SOAP web service that prints student's name.

F. Expected Affective domain Outcome (ADos)

- Follow coding standards.
- Demonstrate working as a leader/ a team member.
- Follow ethical practices.

G. Prerequisite Theory:

On the World Wide Web, a web service is a standardized method for propagating messages between client and server applications. Web services can be used by software programs written in a variety of programming languages and running on a variety of platforms to exchange data via computer networks such as the Internet in a similar way to inter-process communication on a single computer. Any software, application, or cloud technology that uses standardized web protocols (HTTP or HTTPS) to connect, interoperate, and exchange data messages – commonly XML (Extensible Markup Language) – across the internet is considered a web service.

Web services have the advantage of allowing programs developed in different languages to connect with one another by exchanging data over a web service between clients and servers. A client invokes a web service by submitting an XML request, which the service responds with an XML response.

SOAP stands for "Simple Object Access Protocol." It is a transport-independent messaging protocol. SOAP is built on sending XML data in the form of SOAP Messages. A document known as an XML document is attached to each message. Only the structure of the XML document, not the content, follows a pattern. The best thing about Web services and SOAP is that everything is sent through HTTP, the standard web protocol.

A root element known as the element is required in every SOAP document. In an XML document, the root element is the first element. The "envelope" is separated into two halves. The header comes first, followed by the body. The routing data, or information that directs the XML document to which client it should be sent to, is contained in the header. The real message will be in the body.

H. Resources/Equipment Required

Sr. No.	Instrument/Equipment /Components/Trainer kit	Specification	Quantity
1	Hardware: Computer System	Computer (i3-i5 preferable),RAM 2 GB onwards	As per batch size
2	Software: Operating System	Windows/ Linux	
3	Software	Java 8 is required on the Linux, windows or mac operating system. Apache Tomcat 9.0 Eclipse IDE for Java EE Developers (Eclipse Oxygen can be used)	

I. Safety and necessary Precautions followed.

- Shutdown computer system properly once the Lab hours is finished.

J. Procedure to be followed/Source code:

SOAP is based on Web Services Description Language (WSDL).

Make sure your IDE contains Apache Tomcat and Apache Axis. Apache Axis will do the work of creating web service using Java source file and Apache Tomcat server will be used to run and test the web service.

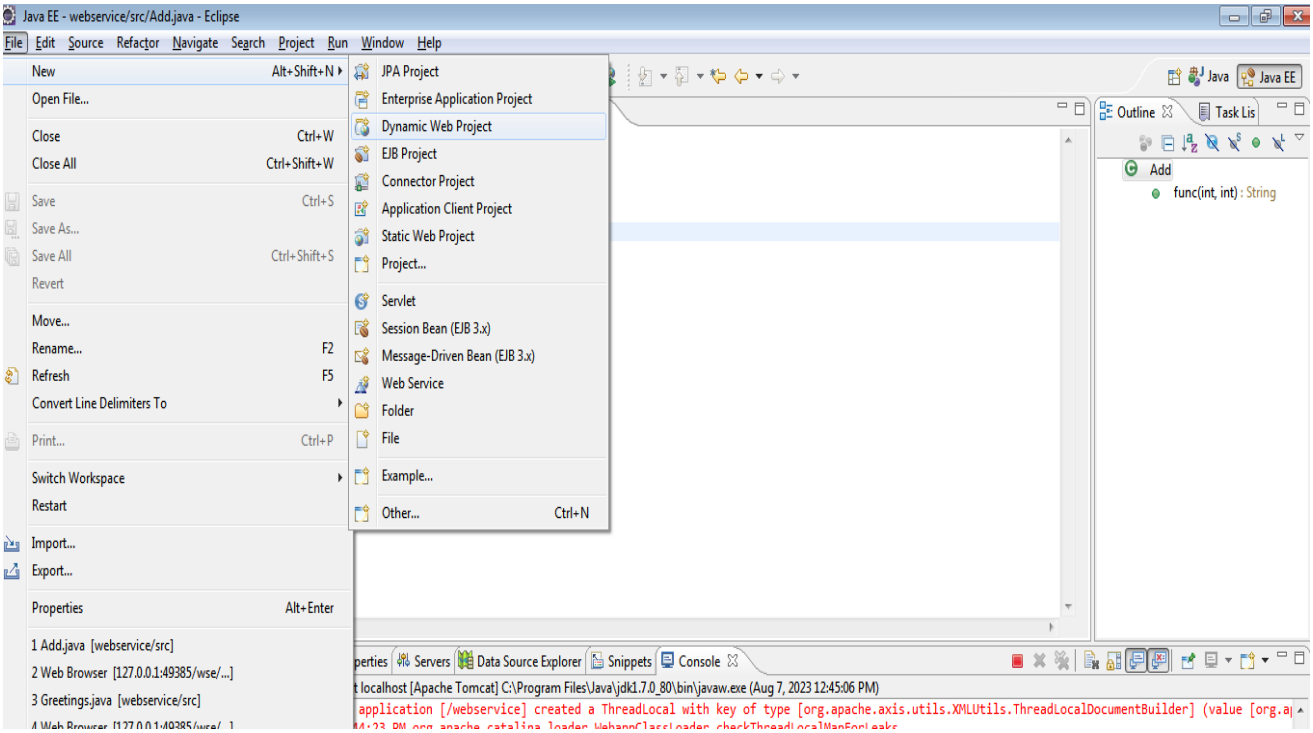
The code below shows the service to add two numbers.

Add.java

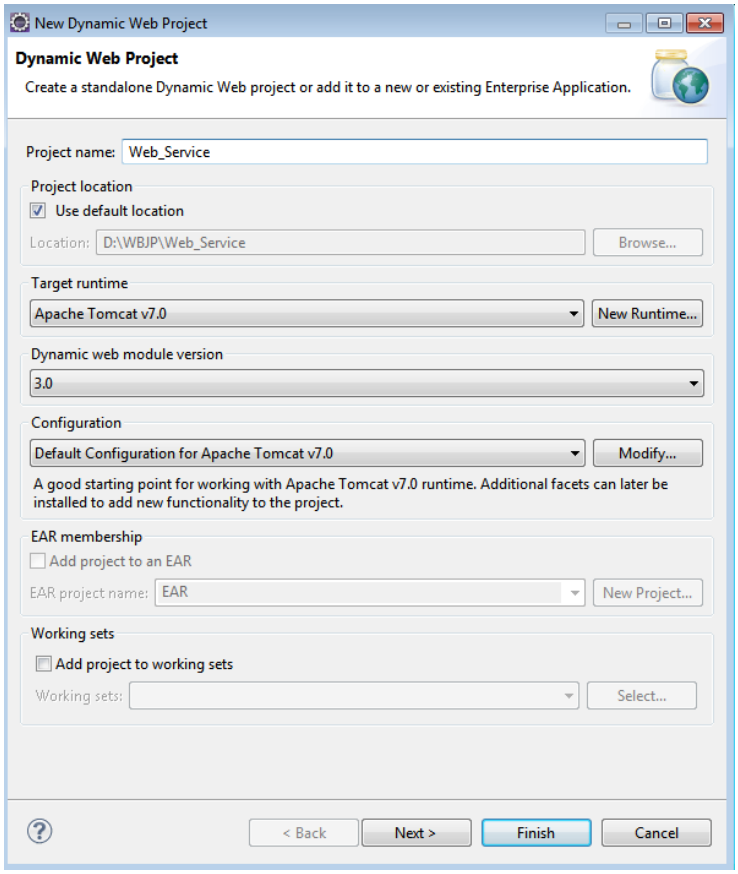
```
public class Add {
        public String func(int a,int b){
                return String.valueOf(a+b);
                }
}
```

Follow the procedure demonstrated below

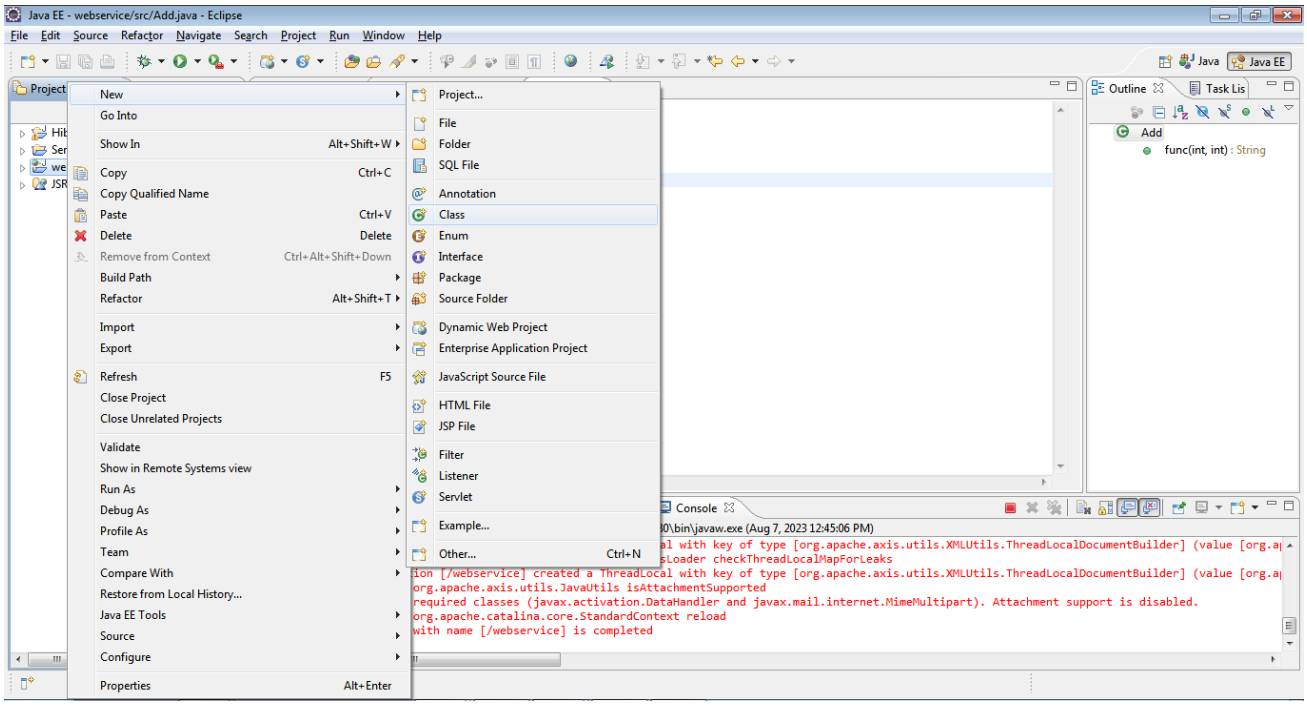
- First of all open Eclipse and go to File > New > Dynamic Web Project



- Give a project name and then click on Finish button to create a dynamic web project.

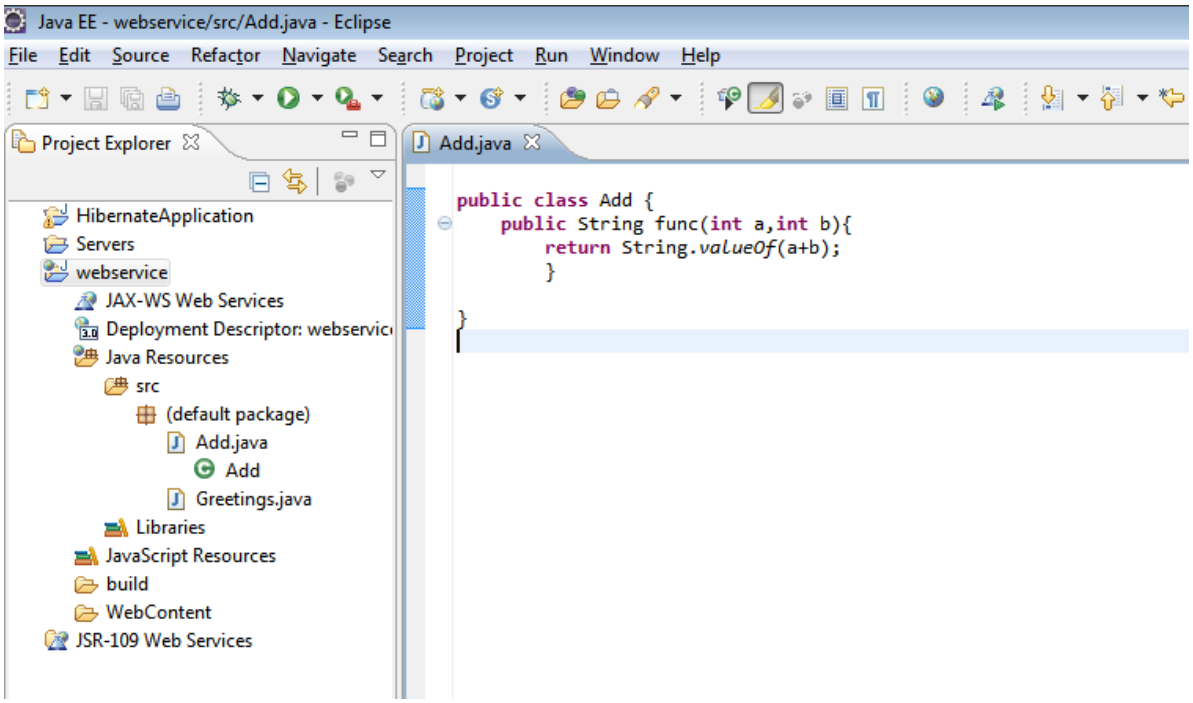


- Now create a Java class inside src folder

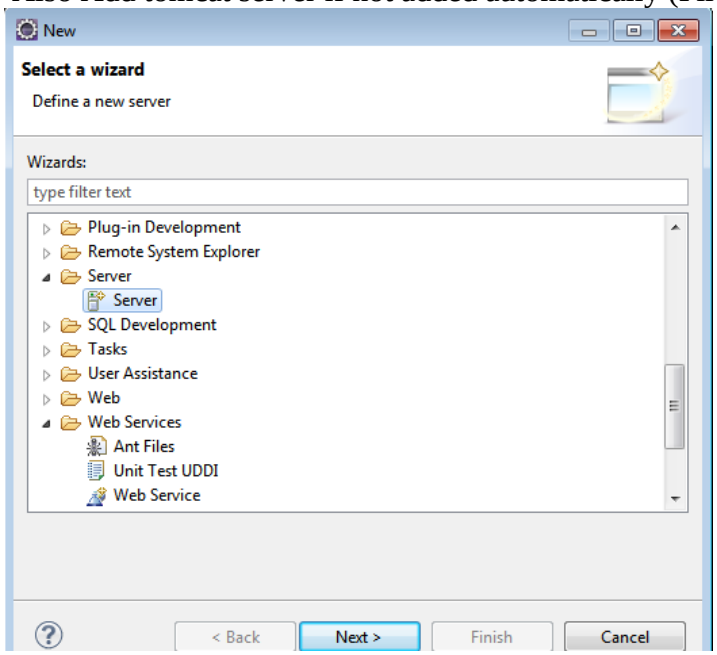


- For demonstration purpose creates a function in the class that will take two numbers and return sum of these two numbers. Add following code in your class. You can create more functions according to your requirement.

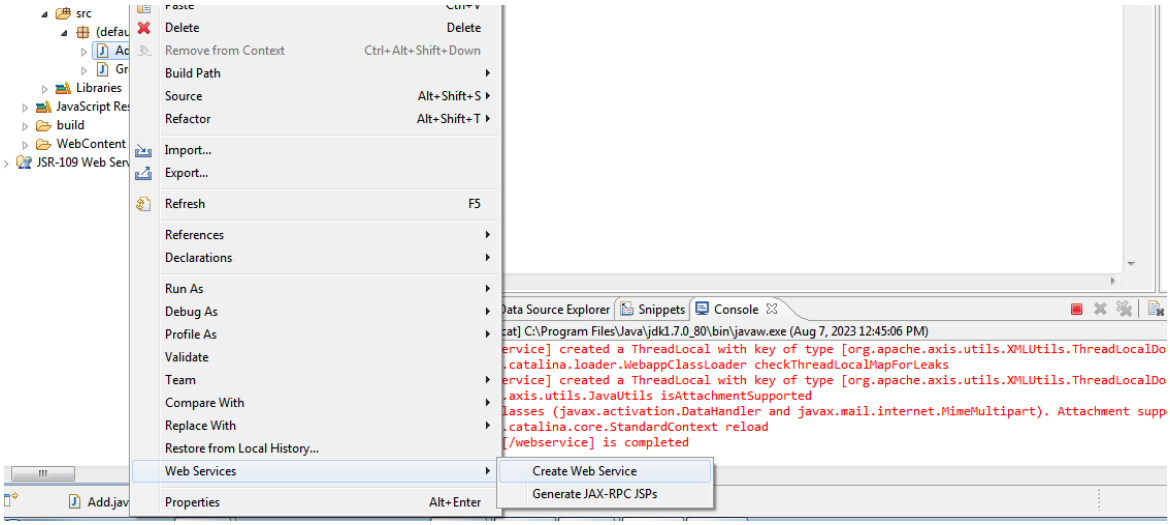
```
public class Add {
        public String func(int a,int b){
                return String.valueOf(a+b);
                }

}
```



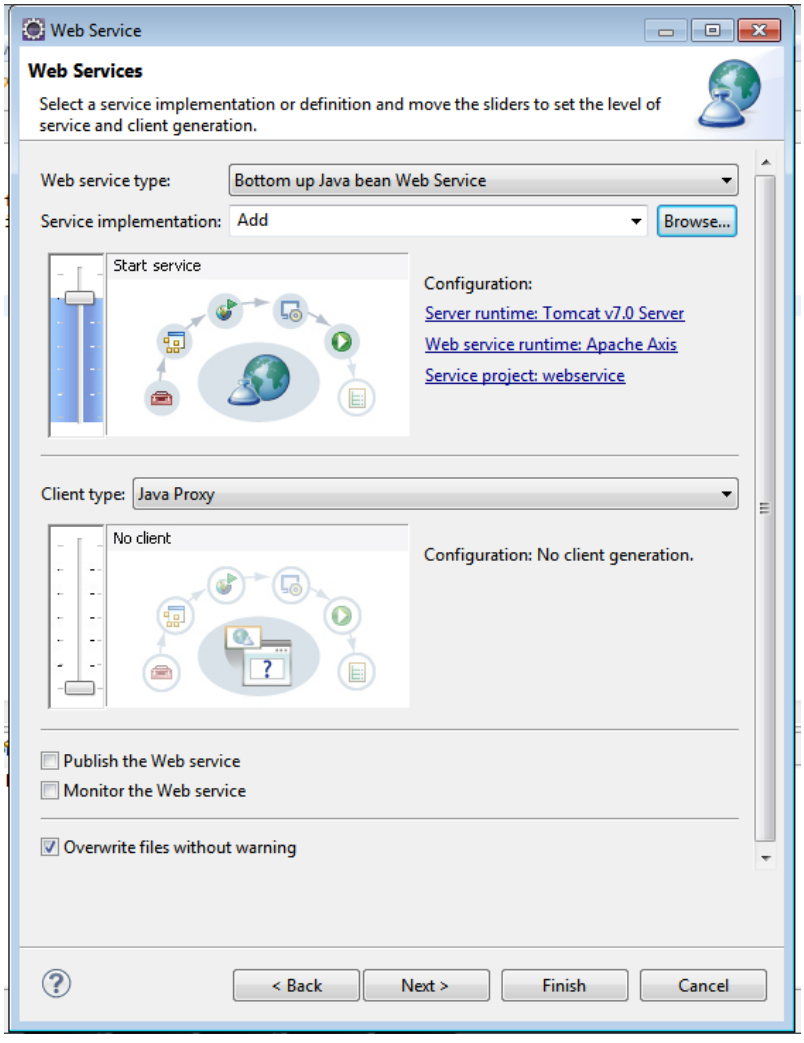
- Also Add tomcat server if not added automatically (File → New → Server)



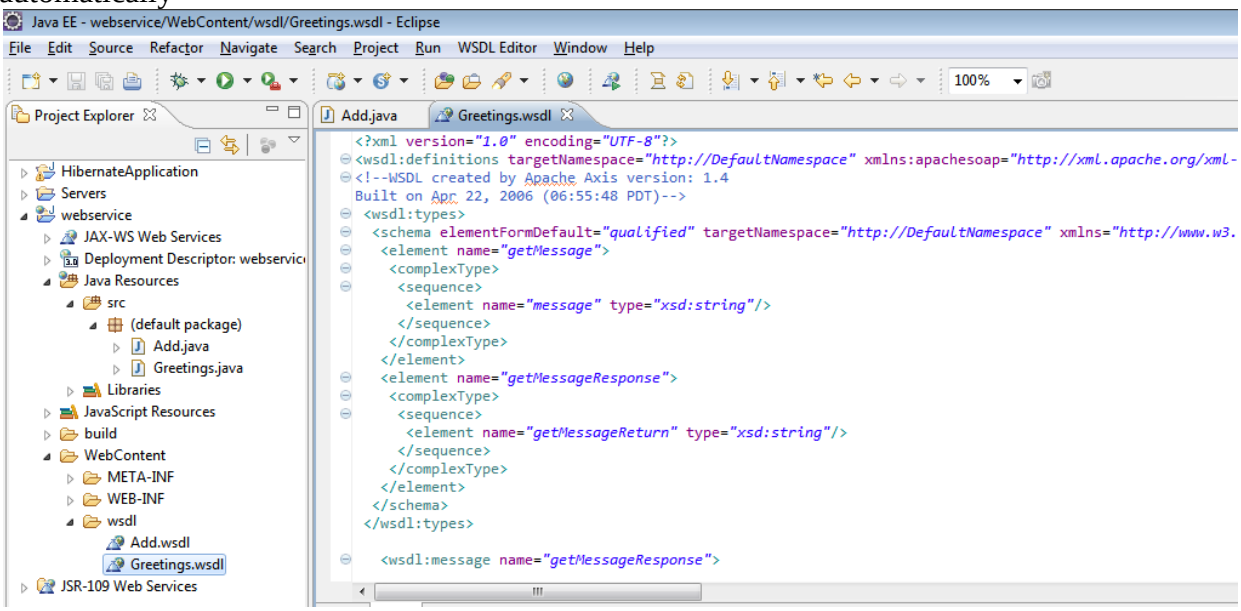
- Right click on Java class and go to Web Services > Create Web Service.



- Select all the settings as given in below screenshot. Finally click on Finish button to create the web service.



- Under project explorer Go to your project - > WebContent -> wsdl -> Add.wsdl is generated automatically

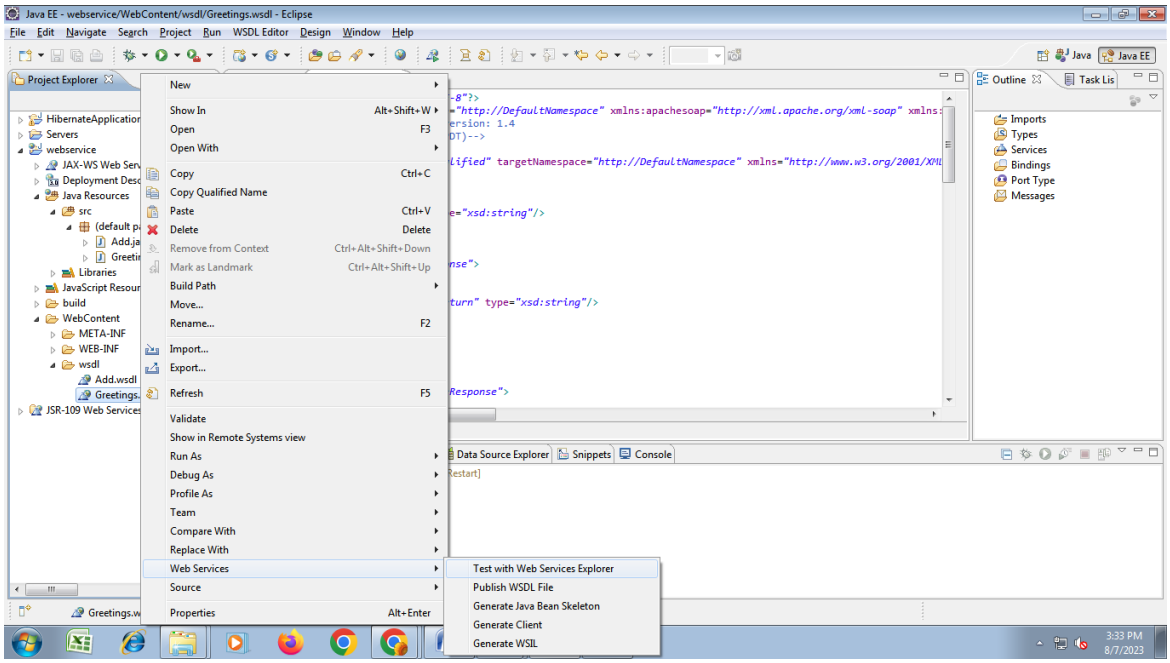


Add.wsdl

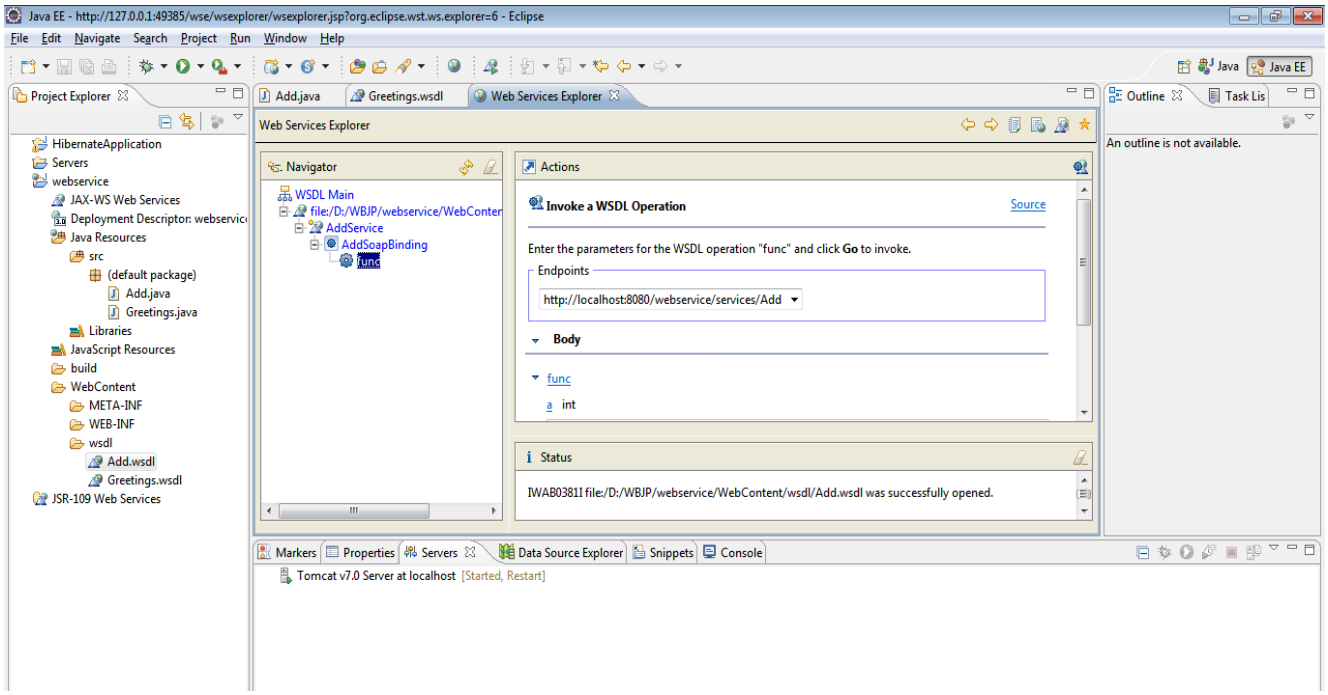
```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://DefaultNamespace"
xmlns:apachesoap="http://xml.apache.org/xml-soap" xmlns:impl="http://DefaultNamespace"
xmlns:intf="http://DefaultNamespace" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<!--WSDL created by Apache Axis version: 1.4
Built on Apr 22, 2006 (06:55:48 PDT)-->

 <wsdl:types>
  <schema elementFormDefault="qualified" targetNamespace="http://DefaultNamespace"
xmlns="http://www.w3.org/2001/XMLSchema">

   <element name="func">
    <complexType>
     <sequence>
      <element name="a" type="xsd:int"/>
      <element name="b" type="xsd:int"/>
     </sequence>
    </complexType>
   </element>

   <element name="funcResponse">
    <complexType>
     <sequence>
      <element name="funcReturn" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
```

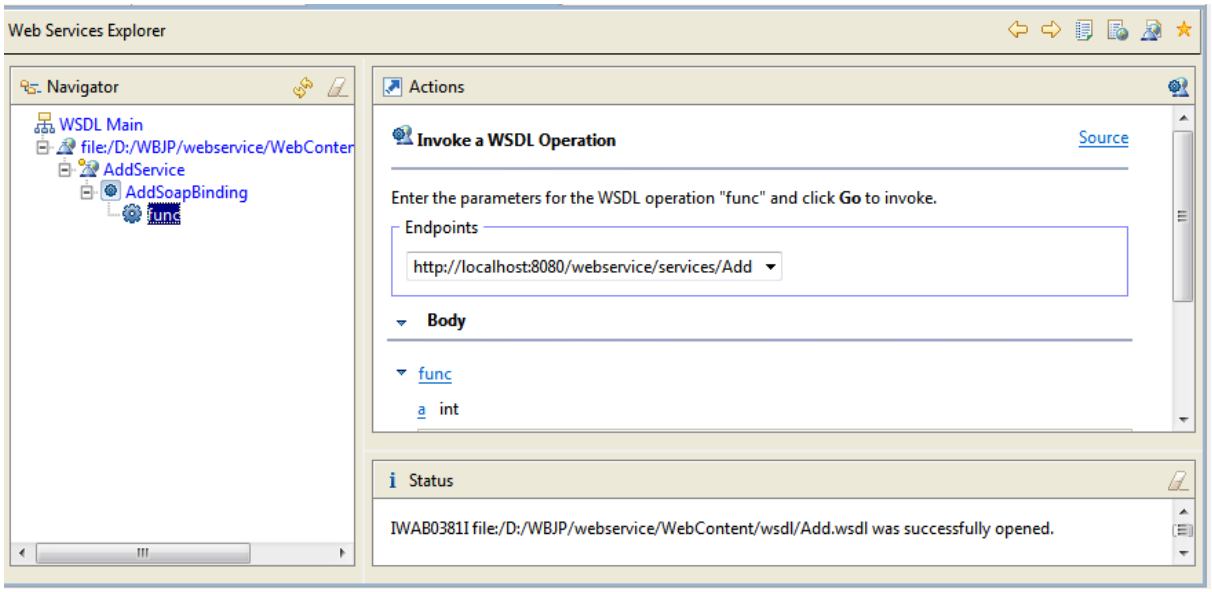
```
 </schema>

</wsdl:types>
  <wsdl:message name="funcRequest">
    <wsdl:part element="impl:func" name="parameters">
    </wsdl:part>
  </wsdl:message>

  <wsdl:message name="funcResponse">
    <wsdl:part element="impl:funcResponse" name="parameters">
    </wsdl:part>
  </wsdl:message>

  <wsdl:portType name="Add">
    <wsdl:operation name="func">
      <wsdl:input message="impl:funcRequest" name="funcRequest">
      </wsdl:input>


      <wsdl:output message="impl:funcResponse" name="funcResponse">
      </wsdl:output>
    </wsdl:operation>
  </wsdl:portType>

  <wsdl:binding name="AddSoapBinding" type="impl:Add">
    <wsdlsoap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="func">
      <wsdlsoap:operation soapAction=""/>

      <wsdl:input name="funcRequest">
        <wsdlsoap:body use="literal"/>
      </wsdl:input>

      <wsdl:output name="funcResponse">
        <wsdlsoap:body use="literal"/>
      </wsdl:output>

     </wsdl:operation>
   </wsdl:binding>

  <wsdl:service name="AddService">
    <wsdl:port binding="impl:AddSoapBinding" name="Add">
      <wsdlsoap:address location="http://localhost:8080/webservice/services/Add"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

- Now to run above web service under project explorer Go to your project - > WebContent -> wsdl -> Add.wsdl. Right Click on Add.wsdl file → Choose Web Services → Test with web services explorer

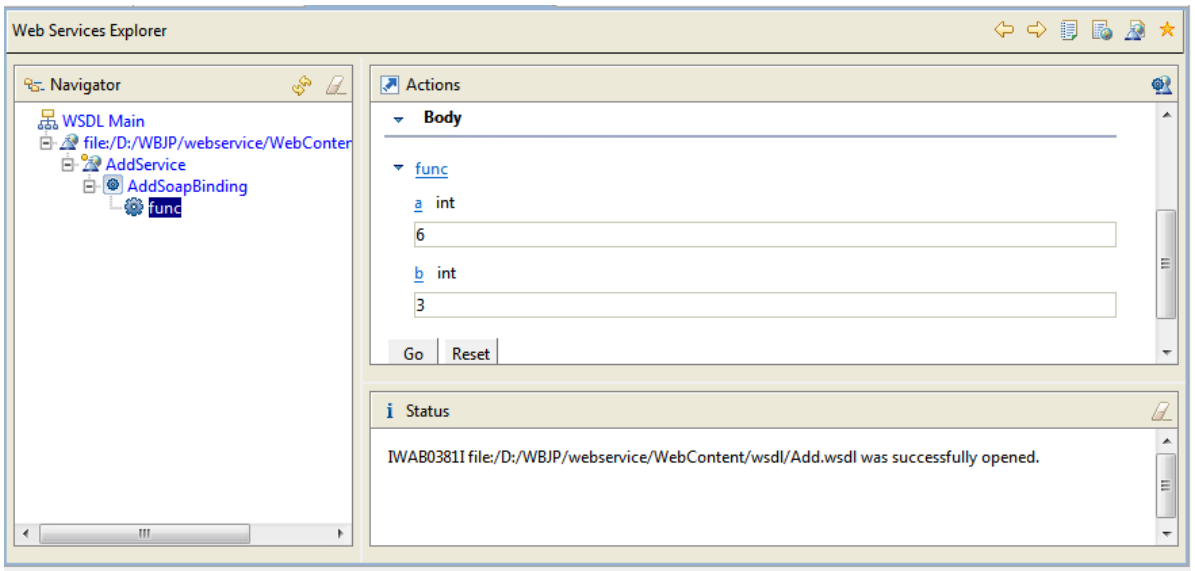


- Web Service Explorer will open as shown below. Click on func() in the Navigator section

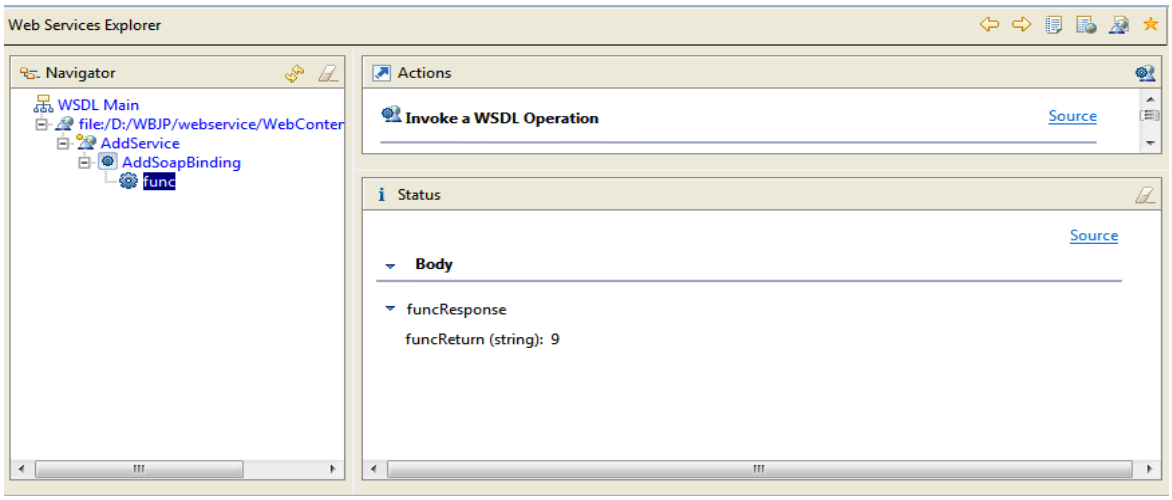




- Under Actions Section → Body add two numbers and click on Go



- Under status section you will now see the output



Now follow the same above procedure and write java file to print your name

……

……

……

……

……

……

……

……

……

……

……

……

K. Input-Output:

L. Practical related Quiz.

a. Why do we need a web service?

b. What are applications of web service?

M. References / Suggestions

- https://www.geeksforgeeks.org/what-are-web-services/
- https://java2blog.com/soap-web-service-example-in-java-using/
- https://examples.javacodegeeks.com/soap-web-service-example-in-java/
- https://www.thecrazyprogrammer.com/2016/02/create-java-soap-web-service-using-eclipse.html

Sign with Date

Web based Java Programming

4350708

Lab manuals are prepared by

Ms. Drashti S Baldev
Lecturer
Government Polytechnic for Girls, Ahmedabad

Ms Jasmine J. Karagthala
Lecturer
Government Polytechnic for Girls, Ahmedabad

Branch Coordinator

Shri B. H. Kantevala

HOD- Diploma in Computer Engineering

Government Polytechnic, Ahmedabad

Committee Chairman

Shri R. D. Raghani

(HOD-EC)

Principal (I/C)

Government Polytechnic, Gandhinagar