Bharati Vidyapeeth's

Institute of Management & Information Technology

C.B.D. Belapur, Navi Mumbai 400614

Vision:
Providing high quality, innovative and value-based education in information technology to build competent professionals.
Mission
M1. Technical Skills:-To provide solid technical foundation theoretically as well as practically capable of providing quality services to industry.
M2. Development: -Department caters to the needs of students through comprehensive educational programs and promotes lifelong learning in the field of computer Applications.
M3. Ethical leadership:-Department develops ethical leadership insight in the students to succeed in industry, government and academia.

CERTIFICATE

This is to certify that the journal is the work of Mr. / Ms. **PRANALI RAMCHANDRA PALVE** Roll No. **36** of MCA (Sem- 1 Div: B) for the academic year 2022 - 2023

Subject Code: MCAL12

Subject Name: Advanced Java Lab

Subject-in-charge Principal

Date: _____

External Examiner

Date: _____

Problem Statement 1 : Write a Java Program to demonstrate a Generic Class.

Code :

```
class geg<T>
{
   T obj;
   geg(T obj){this.obj = obj;}
   public T get() {return this.obj;}
}
class G1
{
   public static void main (String[] args)
   {
      geg<Integer>i=new geg<Integer>(35);
      System.out.println(i.get());

      geg<String> s =
      new geg<String>(“Pranali");
      System.out.println(s.get());
   }
}
```

Output :

```
Problems  @ Javadoc  Declaration  Console
<terminated> G1 [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (24-Oct-2024, 9:46:04 pm)
35
Pranali
```

Problem Statement 2 : Write a Java Program to demonstrate Generic Methods.

Code :

```
public class Genericmethod
{
   void display()
   {
      System.out.println("generic method exmaple");
   }
   <T> void gdisplay (T e)
   {
      System.out.println(e.getClass().getName() + " = " + e);
   }
   public static void main(String[] args)
   {
      Genericmethod g1=new Genericmethod();
      g1.display();

      g1.gdisplay(1);
      g1.gdisplay("Pranali");
      g1.gdisplay(12.0);
   }
   }
```

Output :

Problem Statement 3 : Write a Java Program to demonstrate Wildcards in Java Generics.

Code :

```
import java.util.*;
public class Wildcards {
   // Upper bounded
   private static double sum(List<? Extends Number> list) {double sum =
       0.0;
       for (Number i : list) {
           sum = sum + i.doubleValue();
       }
       return sum;
   }
   // Lower Bounded
   private static void show(List<? Super Integer> list) {
       list.forEach((x) -> {
 System.out.print(x + " ");
       });
   }
    public static void main(String[] args) {
          System.out.println("Upper Bounded : ");
         List<Integer> list1 = Arrays.asList(4, 2, 7, 5, 1, 9);
         System.out.println("List 1 Sum : " + sum(list1));
         List<Double> list2 = Arrays.asList(4.7, 2.4, 7.3, 5.4, 1.5, 9.2);
        System.out.println("List 2 Sum : " + sum(list2));
        System.out.println("\nLower Bounded : ");
        List<Integer> list3 = Arrays.asList(4, 2, 7, 5, 1, 9);
       System.out.println("Only Classes With Integer Superclass will be Accepted : ");
       show(list3);
   }
}
```

Output :

```
Problems   Javadoc   Declaration   Console
<terminated> Wildcards [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (24-Oct-2024, 10:27:49 pm)
Upper Bounded :
List 1 Sum : 28.0
List 2 Sum : 30.499999999999996

Lower Bounded :
Only Classes With Integer Superclass will be Accepted :
4 2 7 5 1 9
```

Assignment 2

List Interface

1. Write a Java program to create List containing list of items of type String and use for- --each loop to print the items of the list.
2. Write a Java program to create List containing list of items and use ListIterator interface to print items present in the list. Also print the list in reverse/ backward direction.

Problem Statement 1 : Write a Java program to create List containing list of items of type String and use for- --each loop to print the items of the list.

Code :

```
package list;
import java.util.*;
public class Array{
        public static void main(String[] args) { ArrayList<String>list=new
                ArrayList<String>();

                list.add("DMBA");
                list.add("ADBMS");
                list.add("JAVA");
                list.add("WT");

                System.out.println(list);

                System.out.println("Traversing list through for each loop");for(String
                subject:list)
                        System.out.println(subject);
  }
}
```

Output :

```
Problems  @ Javadoc  Declaration  Console
<terminated> Array [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (24-Oct-2024, 10:35:04 pm)
[DMBA, ADBMS, JAVA, WT]
Traversing list through for each loop
DMBA
ADBMS
JAVA
WT
```

\

Problem Statement 2 : Write a Java program to create List containing list of items and use ListIterator interface to print items present in the list. Also print the list in reverse/ backward direction.

Code :

```
package list;
 import java.util.*;
public class Reverse {

        public static void main(String[] args) {
             List<String> mylist = new ArrayList<String>();

            mylist.add("Pranali");
            mylist.add("Nidhi");
            mylist.add("Piyusha");
            mylist.add("Pradnya");
            mylist.add("Vidhi");

           System.out.println("Traversing through iterator");
           System.out.println("Original List:");
           Iterator itr=mylist.iterator();
           while(itr.hasNext()) {
                  System.out.println(itr.next());
           }
           Collections.reverse(mylist);
           System.out.println(); //space between two lines
           System.out.println("Reversed List:");
           Iterator itr1=mylist.iterator();
           while(itr1.hasNext()) {

                   System.out.println(itr1.next());
          }


      }

}
```

Output :

```
Problems  @ Javadoc  Declaration  Console
<terminated> Reverse [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (24-Oct-2024, 10:51:32 pm)
Traversing through iterator
Original List:
Pranali
Nidhi
Piyusha
Pradnya
Vidhi

Reversed List:
Vidhi
Pradnya
Piyusha
Nidhi
Pranali
```

Assignment 3

Set Interface

1. Write a Java program to create a Set containing list of items of type String and print the items in the list using Iterator interface. Also print the list in reverse/ backword direction.
2. Write a Java program using Set interface containing list of items and perform the following operations:
a. Add items in the set.
b. Insert items of one set in to other set.
c. Remove items from the set
d. Search the specified item in the set

Problem Statement 1 : Write a Java program to create a Set containing list of items of type String and print the items in the list using Iterator interface. Also print the list in reverse/ backword direction.

Solution :

```
import java.util.*;
public class Reverse {
public static void main(String[] args) {
// Let us create a list of strings
List<String> mylist = new ArrayList<String>();
mylist.add("Pranali");
mylist.add("Nidhi");
mylist.add("Piyusha");
 mylist.add("Pradnya");
 System.out.println("Original list ");

Iterator<String> itr=mylist.iterator();//getting the Iterator
while(itr.hasNext()){//check if iterator has the elements
 System.out.println(itr.next());
}
Collections.reverse(mylist);
System.out.println(" ");
System.out.println("reversed list ");

Iterator<String> itr1=mylist.iterator();//getting the Iterator
while(itr1.hasNext()){//check if iterator has the elements

System.out.println(itr1.next());
}
}
}
```

Output :

```
Problems  @ Javadoc  Declaration  Console
<terminated> Reverse (1) [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (24-Oct-2024, 11:59:09 pm)
Original list
Pranali
Nidhi
Piyusha
Pradnya

reversed list
Pradnya
Piyusha
Nidhi
Pranali
```

Problem Statement2 : Write a Java program using Set interface containing list of items and perform the following operations:

a. Add items in the set.
b. Insert items of one set in to other set.
c. Remove items from the set
d. Search the specified item in the set

Solution :

```
import java.util.*;
public class set2{
public static void main(String[] args) {
// TODO Auto-generated method stub
Set<Integer> s = new LinkedHashSet<Integer>();
s.add(69);
s.add(99);
s.add(18);
s.add(78);
s.add(97);
s.add(156);
Set<Integer> s1 = new LinkedHashSet<Integer>();
s1.add(50);
s1.add(85);

s.addAll(s1);
System.out.println("Set1: " + s);
System.out.println("Set2: " + s1);
System.out.println();
System.out.println("After Adding set2 into set1: " + s);
s.remove(18);
s.remove(78);
System.out.println("Set after removing elements: " + s);
System.out.println();
System.out.println("Does the Set contains: 99? "
+ s.contains(99));
System.out.println("Does the Set contains: 86? "
+ s.contains(86));
}
```

Output :

Problems @ Javadoc Declaration Console

<terminated> set2 [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (25-Oct-2024, 12:13:28 am)

```
Set1: [69, 99, 18, 78, 97, 156, 50, 85]
Set2: [50, 85]

After Adding set2 into set1: [69, 99, 18, 78, 97, 156, 50, 85]
Set after removing elements: [69, 99, 97, 156, 50, 85]

Does the Set contains: 99? true
Does the Set contains: 86? false
```

Assignment 4

Map Interface

1. **Write a Java program using Map interface containing list of items having keys and associated values and perform the following operations:**

a. **Add items in the map.**
b. **Remove items from the map**
c. **Search specific key from the map**
d. **Get value of the specified key**
e. **Insert map elements of one map in to other map.**
f. **Print all keys and values of the map.**
g. **Write a Java program using Map interface containing list of items having keys and associated values and perform the following operations:**

```
import java.util.*;

public class mapinterface1 {
    public static void main(String[] args) {
        // Declare and initialize map
        Map<Integer, String> map = new HashMap<>();
        map.put(1, "Pranali");
        map.put(2, "Pradnya");
        map.put(3, "Nidhi");
        map.put(4, "Piyusha");
        map.put(5, "Vidhi");

        System.out.println("Map 1:");
        for (Map.Entry<Integer, String> e : map.entrySet()) {
            System.out.println(e.getKey() + " " + e.getValue());
        }
        System.out.println();

        // Declare and initialize second map
        Map<Integer, String> map1 = new HashMap<>();
        map1.put(6, "Shruti");
        map1.put(7, "Sakshi");
        map1.put(8, "Kaustubh");

        System.out.println("Map 2:");
        for (Map.Entry<Integer, String> e : map1.entrySet()) {
            System.out.println(e.getKey() + " " + e.getValue());
        }
        System.out.println();

        // Insert map1 into map2
        System.out.println("Insert map into another map");
        Map<Integer, String> map2 = new HashMap<>();
        map2.putAll(map);
        map2.putAll(map1);
        System.out.println(map2);
        System.out.println();
```

```
        // Remove an item from the map
        System.out.println("Remove items from the map");
        map.remove(3); // Removes the entry with key 3
        for (Map.Entry<Integer, String> e : map.entrySet()) {
            System.out.println(e.getKey() + " " + e.getValue());
        }
        System.out.println();

        // Search for a specific key in the map
        System.out.println("Search specific key from the map");
        System.out.println("Is the key '2' present? " + map.containsKey(2));
        System.out.println("Is the key '6' present? " + map.containsKey(6)); // map1 has key '6', but not map

        System.out.println();

        // Get value of a specific key
        System.out.println("Get value of the specified key");
        String val = map.get(2); // Get the value for key '2'
        System.out.println(val);
        System.out.println();
    }
}
```

Output :

```
Problems  @ Javadoc  Declaration  Console
<terminated> mapinterface1 [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (25-Oct-2024, 12:30:14 am)
Map 1:
1 Pranali
2 Pradnya
3 Nidhi
4 Piyusha
5 Vidhi

Map 2:
6 Shruti
7 Sakshi
8 Kaustubh

Insert map into another map
{1=Pranali, 2=Pradnya, 3=Nidhi, 4=Piyusha, 5=Vidhi, 6=Shruti, 7=Sakshi, 8=Kaustubh}

Remove items from the map
1 Pranali
2 Pradnya
4 Piyusha
5 Vidhi

Search specific key from the map
Is the key '2' present? true
Is the key '6' present? false

Get value of the specified key
Pradnya
```

Assignment 5

Lambda Expressions

1. **Write a Java program using Lambda Expression to print “Hello World!”.**
2. **Write a Java program using Lambda Expression with single parameter.**
3. **Write a Java program using Lambda Expression with multiple parameters to add two numbers.**
4. **Write a Java program using Lambda Expression to calculate the following:**
    a. **Convert Fahrenheit to Celcius**
    b. **Convert Kilometers to Miles.**
5. **Write a Java program using Lambda Expression with or without return keyword.**
6. **Write a Java program using Lambda Expression to concatenate two strings.**

Problem Statement 1 : Write a Java program using Lambda Expression to print “Hello World!”.

Solution :

```
package Lambdaexpression;

inerface HelloWorld1 {
        String sayHello(String name);
  }
  public class HelloWorld {
        public static void main(String args[]){
                HelloWorld1 helloWorld = (String name) -> {
                return "Hello " + name; };
                 System.out.println(helloWorld.sayHello("World"));
        }

  }
```

Output :

```
Problems  Javadoc  Declaration  Console
<terminated> mapinterface1 [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (25-Oct-2024, 12:58:24 am)
Hello World
```

Problem Statement 2 : Write a Java program using Lambda Expression with single parameter.

Solution :

```
package Lambdaexpression;
interface Say{
  public String say(String name);
}
public class singleparameter{
   public static void main(String[] args) {
        Say s1=(name)->{
        return "Hello "+name;
      };
      System.out.println(s1.say("Pranali"));
   }
}
```

Output :

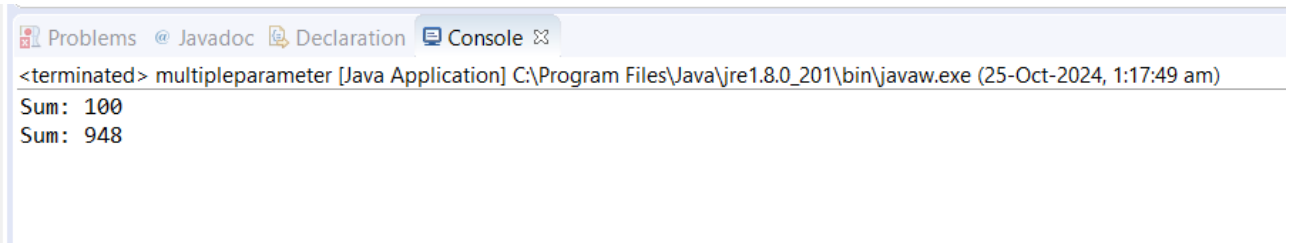
```
Problems  Javadoc  Declaration  Console
<terminated> singleparameter [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (25-Oct-2024, 1:03:58 am)
Hello Pranali
```

Problem Statement 3 : Write a Java program using Lambda Expression with multiple parameters to add two numbers.

Solution :

```
package Lambdaexpression;
interface Add{
   int add(int a,int b);
}

public class multipleparameter{
      public static void main(String[] args) {

           Add ad1=(a,b)->(a+b);
           System.out.println("Sum: " +ad1.add(70,30));

           Add ad2 =(int a,int b)->(a+b);
          System.out.println("Sum: " +ad2.add(709,239));
      }

}
```

Output :



Problem Statement 4 : Write a Java program using Lambda Expression to calculate the following:

a. Convert Fahrenheit to Celsius

Solution :

```
package Lambdaexpression;
interface temp
{
    public double convert(double temp);

}
public class farhernt1 {
        public static void main(String[] args){
                temp t1=(double a)->{
                        return((a-32)* 5/9);
                };
                System.out.print("Convert fahrenheit to celsius: "+ t1.convert(99));
        }


}
```

Output :

```
Problems  @ Javadoc  Declaration  Console
<terminated> farhernt1 [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (25-Oct-2024, 1:21:57 am)
Convert fahrenheit to celsius: 37.22222222222222
```

b. Convert Kilometers to Miles.

Solution :

```
package Lambdaexpression;
 interface temp1
{
public double convert(double temp);

}
public class kmtomiles {
public static void main(String[] args) { temp t1=(double a)->{
return(a/1.6);
};
System.out.print("Convert KM to MILES: "+ t1.convert(10)+ " Miles");
}


}
```

Output :

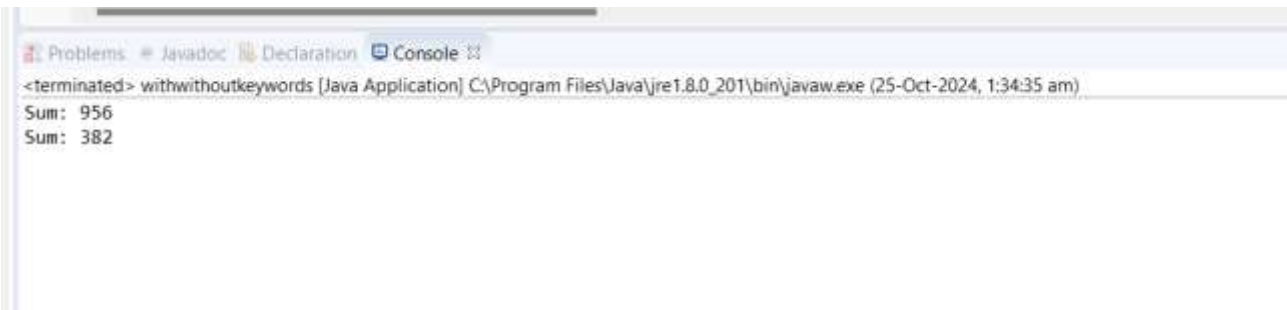
```
Problems  @ Javadoc  Declaration  Console
<terminated> kmtomiles [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (25-Oct-2024, 1:29:29 am)
Convert KM to MILES: 6.25 Miles
```

Problem Statement 5 : Write a Java program using Lambda Expression with or without return keyword.

Solution :

```
package Lambdaexpression;
interface Add2{
    int add(int a,int b);
}

public class withwithoutkeywords {
     public static void main(String[] args) {

          // without return keyword
         Add2 ad1=(a,b)->(a+b);
        System.out.println("Sum: " +ad1.add(78,878));

         // with return keyword
         Add2 ad2=(int a,int b)->
        {
        return (a+b);
         };
         System.out.println("Sum: " +ad2.add(57,325));
    }
}
```

Output :

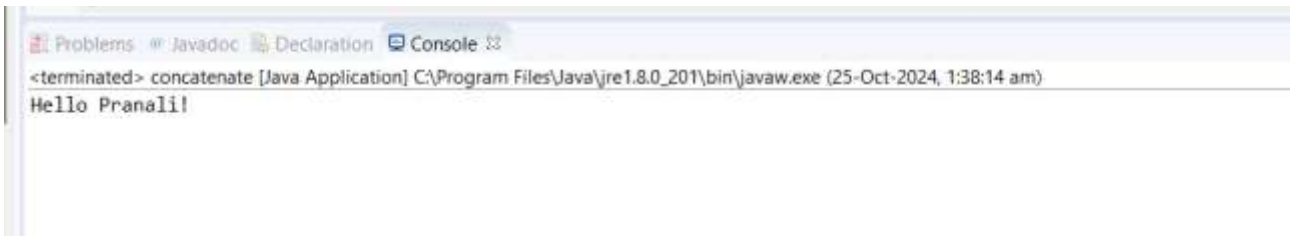


Problem Statement 6 : Write a Java program using Lambda Expression to concatenate two strings.

Solution :

```
package Lambdaexpression;

interface conc1 {
        public String concat(String a,String b);
}
public class concatenate {

        public static void main(String[] args) {
                conc1 s1 = (String a,String b)->{
                        return (a+b);
        };
        System.out.println(s1.concat("Hello"," Pranali!"));
        }
}
```

Output :



Assignments 6

Web Application Development using JSP

1. **Create a Telephone directory using JSP and store all the information within a database, so that later could be retrieved as per the requirement. Make your own assumptions.**

2. **Write a JSP page to display the Registration form (Make your own assumptions)**

3. **Write a JSP program to add, delete and display the records from StudentMaster (RollNo, Name, Semester, Course) table.**

4. **Design loan calculator using JSP which accepts Period of Time (in years) and Principal Loan Amount. Display the payment amount for each loan and then list the loan balance and interest paid for each payment over the term of the loan for the following time period and interest rate:**

a. **1 to 7 year at 5.35%**

b. **8 to 15 year at 5.5%**

c. **16 to 30 year at 5.75%**

5. **Write a program using JSP that displays a webpage consisting Application form for change of Study Center which can be filled by any student who wants to change his/ her study center.**
Make necessary assumptions

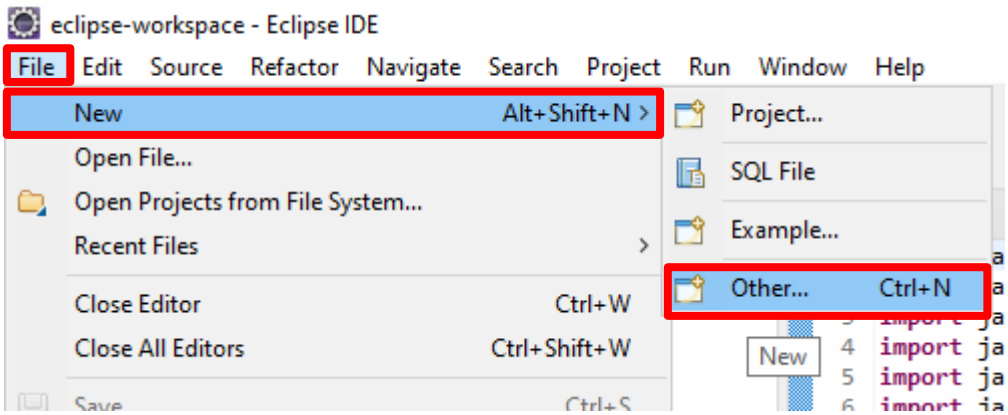
6. **Write a JSP program that demonstrates the use of JSP declaration, scriptlet, directives, expression, header and footer.**

7. **Write a JSP program that demonstrates the use of session.**

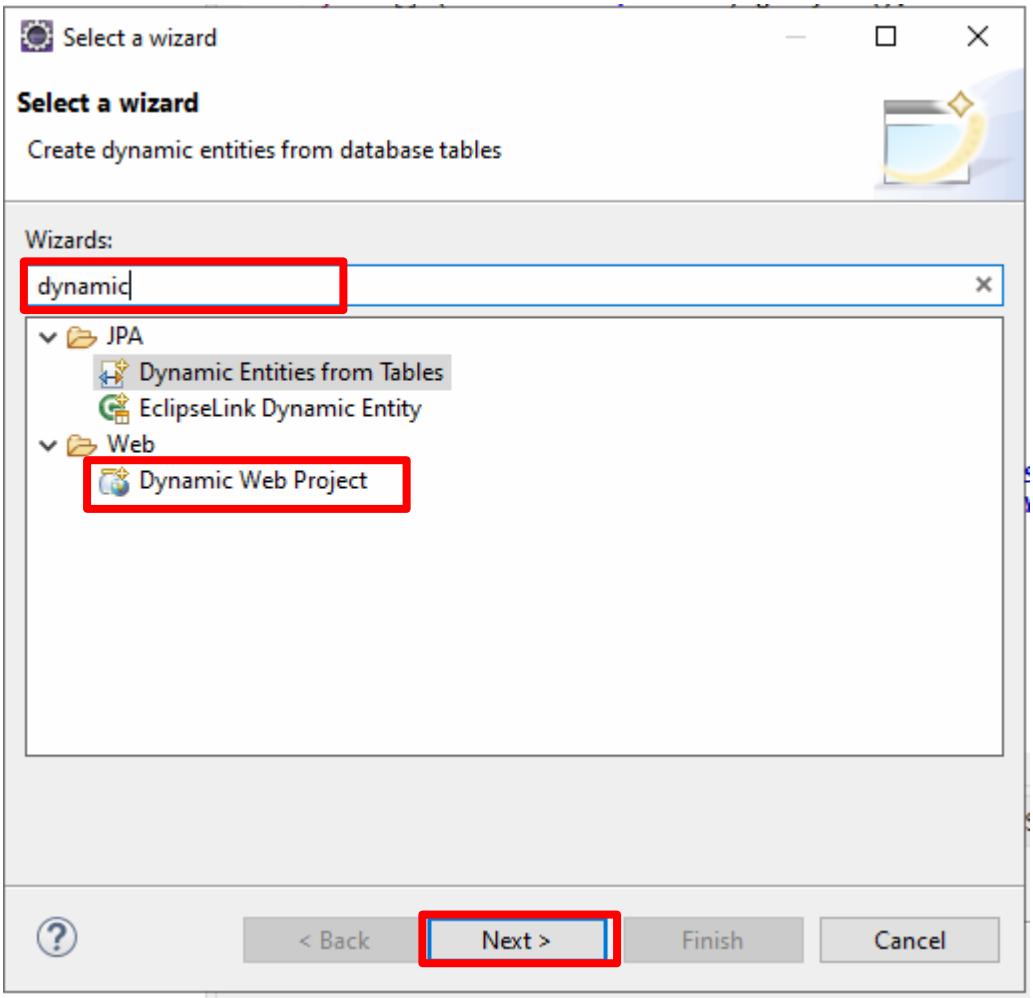
Steps to create Dynamic Web Project

Step 1: Create a new Dynamic Web Project

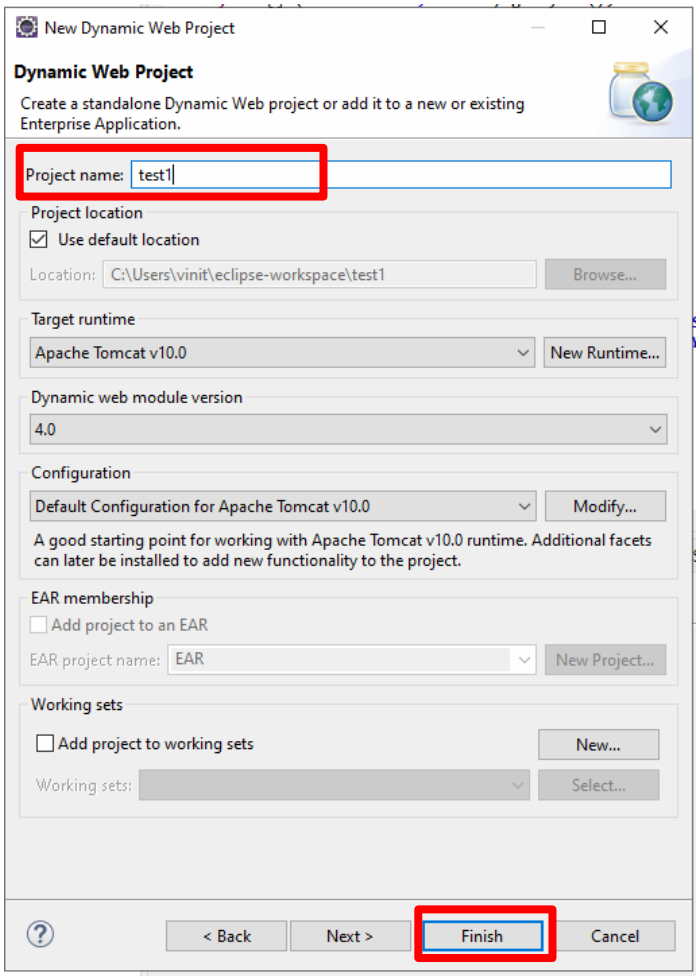
1.1. Click on File – New - Other



1.2. Search for 'Dyanmic' and Select 'Dynamic Web Project'. Then Click on Next



1.3. Enter Project Name of your wish, and click on Finish.



This creates your Dynamic Web project.

a. Create a Telephone directory using JSP and store all the information within a database

Input:

index.jsp

```
<%@page import="java.sql.*"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-
8859-1"%>
<!DOCTYPE html>
<html><head><meta charset="ISO-8859-1"><title>Index</title>
<!-- CSS only -->
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.4.0/font/bootstrap-
icons.css">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta2/dist/css/bootstrap.min.css"rel="stylesheet" integrity="sha384-
BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlolflfl"crossorigin="a
nonymous">
<!-- JavaScript Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
b5kHyXgcpbZJO/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S9FOnJ0"crossorigi
n="anonymous"></script>
</head>
<body>
		<nav class="navbar navbar-dark bg-dark p-4">
			<a class="navbar-brand mb-0 h1">BVIMIT</a>
			<ul class="navbar-nav">
				<li class="nav-item">
				<a class="navbar-link text-light text-decoration-
none"href='add.jsp'>Add Phone</a>
				</li></ul>
		</nav>
		<br>	<br>
		<table class="table table-stripped">
			<tr class="text-center">
				<th>Id</th>
				<th>Name</th>
				<th>Phone</th>
				<th>Delete</th>
			</tr>
		<%
		try{
			String driver ="org.postgresql.Driver";
			String url ="jdbc:postgresql://localhost:5432/postgres";
			String username ="postgres";
```

```
                    String password ="1234";
                Connection con=null;
                Class.forName(driver).newInstance();
                    con = DriverManager.getConnection(url,username,password);
                    System.out.println("Opened database successfully");
                if(request.getParameter("del")!=null){
                    Statement stmt = con.createStatement();
                   stmt.execute("DELETE FROM TeleDir Where id = " +
request.getParameter("del"));
                    response.sendRedirect("index.jsp");
                }
                 String myDataField =null;
                 String myQuery ="SELECT * FROM TeleDir ORDER BY id ASC";
                PreparedStatement myPreparedStatement =null;
                ResultSet myResultSet =null;
                myPreparedStatement = con.prepareStatement(myQuery);
                ResultSet rs = myPreparedStatement.executeQuery();
                while(rs.next()){ %>
                <tr class="text-center">
                    <td><%= rs.getInt(1) %></td>
                    <td><%= rs.getString(2) %></td>
                    <td><%= rs.getString(3) %></td>
                   <td><a href="?del=<%= rs.getInt(1) %>"><i class="bi bi-trash-fill text-
danger"></i></a></td>
                </tr>
                <%
                }
                }catch(Exception e){
                System.out.println(e);
                }
                %>
                </table>

                </body>
                </html>
```

add.jsp

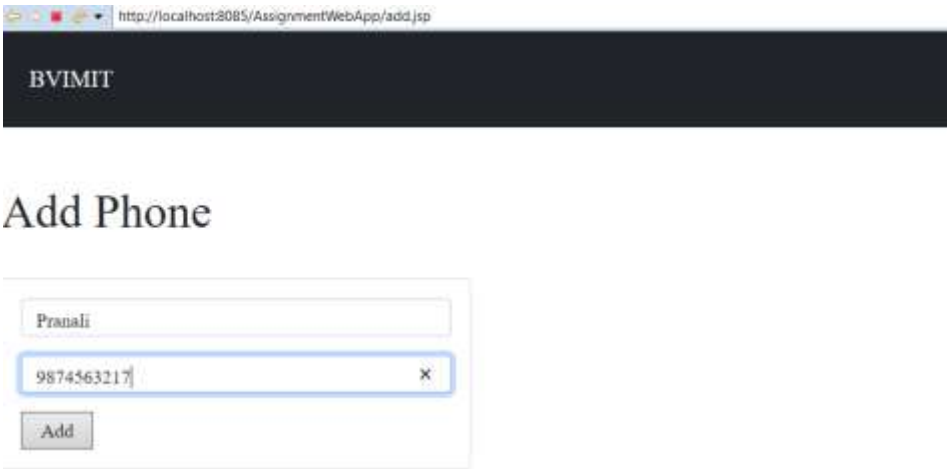
```
 <%@page import="java.sql.*"%>
 <%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-
 8859-1"%>
<!DOCTYPE html>
<html>
<head>

        <meta charset="ISO-8859-1">
```

```
        <title>Add</title>

<!-- CSS only -->
        <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.4.0/font/bootstrap-icons.css">
        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta2/dist/css/bootstrap.min.css"rel="stylesheet" integrity="sha384-
BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlolflfl"crossorigin="anon
ymous">
<!-- JavaScript Bundle with Popper -->
        <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta2/dist/js/bootstrap.bundle.min.js" integrity="sha384-
b5kHyXgcpbZJO/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S9FOnJ0"crossorigin=
"anonymous"></script>
</head>
<body>

                <nav class="navbar navbar-dark bg-dark p-4">
                        <a class="navbar-brand mb-0 h1">BVIMIT</a>
                        <ul class="navbar-nav">
                                <li class="nav-item">
                                        <a class="navbar-link text-light text-decoration-
none"href='index.jsp'>Home</a>
                                </li>
                        </ul>
                </nav>
        <br>    <br>
                <h1>    Add Phone</h1>
                <br>
                <form action="add.jsp" method="post" class="card p-2" style="width: 400px">
                        <div class="form-group m-2">
                                <input class="form-control" name="name" type="text"
placeholder="Name" required="required" />
                        </div>
                        <div class="form-group m-2">
                                <input class="form-control" name="phone" type="text"
placeholder="Phone" required="required" pattern="[0-9]{10,10}" title="Ex. 123654789"/>
                        </div>

<div class="form-group m-2">
                                <input class="btnbtn-primary px-3" type="submit" value="Add"/>
                        </div>

                </form>


                <%
```

```
                    try{
                              String driver ="org.postgresql.Driver";
                              String url ="jdbc:postgresql://localhost:5432/postgres";
                              String username ="postgres";
                              String password ="1234";
                     Connection con=null;
                    Class.forName(driver).newInstance();
                              con = DriverManager.getConnection(url,username,password);
                              if(request.getParameter("phone") != null){
                                        PreparedStatement ps = con.prepareStatement("insert into
TeleDir(name, phone) VALUES(?,?)");
                                        ps.setString(1,
request.getParameter("name").toString().toUpperCase());
                                         ps.setString(2, request.getParameter("phone").toString());
                                         if(ps.executeUpdate() > 0){
                                         %>
                                         <p>Phone Added Successfully.</p>
                                         <%
                                         }else {                                  %>
                                         <p>Failed to Add Phone.</p><%
                                         }          }
                    }catch(Exception e){
                    System.out.println(e);
                    }                    %>
</body>
</html>
```

OUTPUT:

Add Data:



Database :

- CREATE

```
create table TeleDir(
        id  serial PRIMARY KEY,
        name varchar(20),
        phone varchar(14)
);
```

Query Editor Query History

```
1  create table TeleDir(
2  id  serial PRIMARY KEY,
3  name varchar(20),
4  phone varchar(14)
5  )
```

Data Output Explain Messages Notifications

```
CREATE TABLE

Query returned successfully in 94 msec.
```

b. Write a JSP page to display the Registration form (Make your own assumptions).

Input:

register_1.jsp

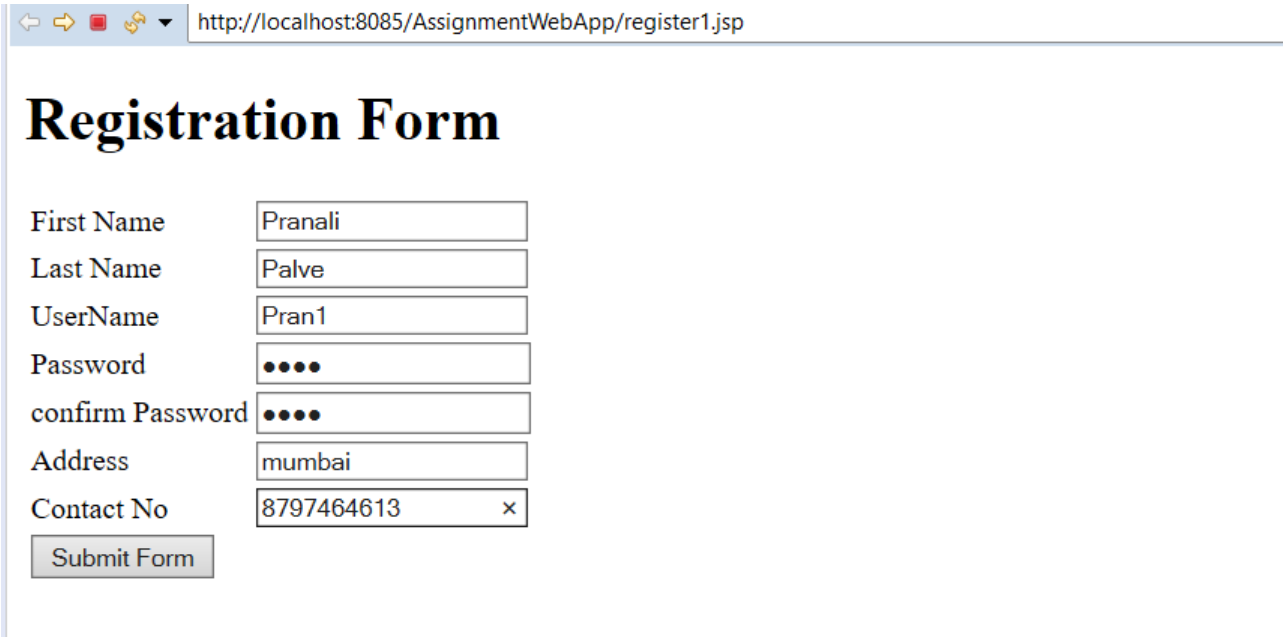
```
<%@ page language="java"contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
 <meta charset="ISO-8859-1">
<title>Registration Form</title>
</head>
<body>
<center>
<h1>Registration Form</h1>

 <form action="register_2.jsp" method="post">
                        <table style="with: 50%">
                                <tr>
                                        <td>First Name</td>
                                        <td><input type="text" name="first_name" /></td>
                                </tr>
                                <tr>
                                        <td>Last Name</td>
                                        <td><input type="text" name="last_name" /></td>
                                </tr>

                                <tr>
                                        <td>UserName</td>
                                        <td><input type="text" name="username" /></td>
                                </tr>
                                        <tr>
                                        <td>Password</td>
                                        <td><input type="password" name="password" /></td>
                                </tr>

                                        <tr>
                                        <td>confirm Password</td>
                                        <td><input type="password" name="cpassword" /></td>
                                </tr>
                                <tr>
                                        <td>Address</td>
                                        <td><input type="text" name="address" /></td>
                                </tr>
                                <tr>
                                        <td>Contact No</td>
```

```
                                        <td><input type="text" name="contact" /></td>
                                    </tr>
                                <tr>
  <td colspan="2"><input type="submit" value="Submit Form" /></td></tr>
</table>
</center>
</body>

</html>
```

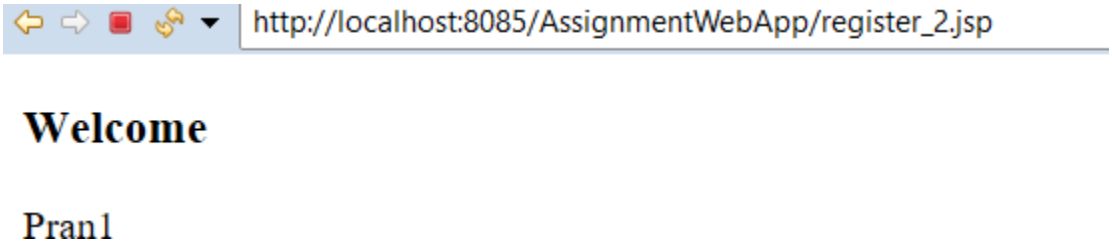
register_2.jsp

```
  <%@ page language="java"contentType="text/html; charset=ISO-8859-1"
      pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
  <meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<%
String n=request.getParameter("username");
String str1=request.getParameter("password");
String str2=request.getParameter("cpassword");
if(str1.equals(str2))

   {
   out.println("<h3>Welcome</h3>"+n);
   }
   else
   {
   out.println("<h3>Sorry, your password is mismatched</h3>");
   }
%>

</body>

</html>
```

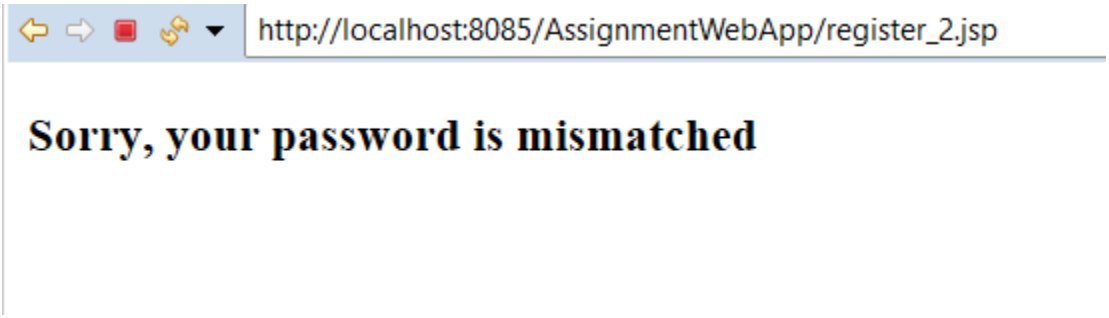
OUTPUT:



After Click on Submit Form



If password is wrong.



c. Write a JSP program to add, delete and display the records from StudentMaster (RollNo, Name, Semester, Course) table.

Input:

Index.jsp

```
<%@page import="java.sql.*"%>
<%@ page language="java"contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Index</title>
<!-- CSS only -->
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.4.0/font/bootstrap-
icons.css">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta2/dist/css/bootstrap.min.css"rel="stylesheet" integrity="sha384-
BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlolflfl"crossorigin="anon
ymous">
<!-- JavaScript Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
b5kHyXgcpbZJO/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S9FOnJ0"crossorigin=
"anonymous"></script>
</head>
<body>
        <center>
                <nav class="navbar navbar-dark bg-dark p-4">
                        <a class="navbar-brand mb-0 h1">BVIMIT</a>
                        <ul class="navbar-nav">
                                <li class="nav-item">
                                        <a class="navbar-link text-light text-decoration-
none"href='add.jsp'>Add Student</a>
                                </li>
                        </ul>
                </nav>
                <br>
                <br>
                <table class="table table-stripped">

                        <tr class="text-center">
                                <th>Id</th>
                                <th>Roll</th>
                                <th>Name</th>
                                <th>Sem</th>

                                <th>Course</th>
```

```
                        <th>Update</th>
                        <th>Delete</th>
                </tr>
        <%
        try{
                String driver ="org.postgresql.Driver";
                String url ="jdbc:postgresql://localhost:5434/postgres";
                String username ="postgres";
                String password ="ravita123";

         Connection con=null;
        Class.forName(driver).newInstance();
                con = DriverManager.getConnection(url,username,password);
                System.out.println("Opened database successfully");

        if(request.getParameter("del")!=null){
                Statement stmt = con.createStatement();
               stmt.execute("DELETE FROM student Where id = " +
request.getParameter("del"));
                response.sendRedirect("index.jsp");
        }

         String myDataField =null;
         String myQuery ="SELECT * FROM student ORDER BY id ASC";

        PreparedStatementmyPreparedStatement =null;
        ResultSetmyResultSet =null;


        myPreparedStatement = con.prepareStatement(myQuery);
        ResultSetrs = myPreparedStatement.executeQuery();

        while(rs.next()){
                %>
        <tr class="text-center">
                <%-- <td><%= rs.getInt(0) %></td> --%>
                <td><%= rs.getInt(1) %></td>
                <td><%= rs.getString(2) %></td>
                <td><%= rs.getString(3) %></td>
                <td><%= rs.getString(4) %></td>
                <td><%= rs.getString(5) %></td>
                <td>
                        <a href="update.jsp?id=<%= rs.getInt(1) %>"><i class="bi bi-
 pencil-fill"></i></a>
               </td>
               <td>

                        <a href="?del=<%= rs.getInt(1) %>"><i class="bi bi-trash-fill text-
```

```
danger"></i></a>
                        </td>
                </tr>
                <%
                }

                }catch(Exception e){
                System.out.println(e);
                }
                %>
                </table>
        </center>
</body>
</html>
```

Filename-add.jsp

```
<%@page import="java.sql.*"%>
<%@ page language="java"contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Add</title>
<!-- CSS only -->
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.4.0/font/bootstrap-
icons.css">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta2/dist/css/bootstrap.min.css"rel="stylesheet" integrity="sha384-
BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlolflfl"crossorigin="anon
ymous">
<!-- JavaScript Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
b5kHyXgcpbZJO/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S9FOnJ0"crossorigin=
"anonymous"></script>
</head>
<body>
        <center>
                <nav class="navbar navbar-dark bg-dark p-4">
                        <a class="navbar-brand mb-0 h1">BVIMIT</a>
                        <ul class="navbar-nav">
                                <li class="nav-item">
                                        <a class="navbar-link text-light text-decoration-none"

href='Index.jsp'>Home</a>
```

```
                        </li>
                    </ul>
                </nav>
            <br>
            <br>
                <h1>
                    Add Student
                </h1>
                <br>

                <form action="add.jsp" method="post" class="card p-2" style="width: 400px">

                    <div class="form-group m-2">
                        <input class="form-control" name="rno" type="text"
 placeholder="Roll No" required="required" />
                    </div>
                    <div class="form-group m-2">
                        <input class="form-control" name="name" type="text"
 placeholder="Name" required="required"/>
                    </div>
                    <div class="form-group m-2">
                        <select class="form-control" name="sem">
                            <option value="Sem1">Semester 1</option>
                            <option value="Sem2">Semester 2</option>
                            <option value="Sem3">Semester 3</option>
                            <option value="Sem4">Semester 4</option>
                            <option value="Sem5">Semester 5</option>
                            <option value="Sem6">Semester 6</option>
                        </select>
                       <!--<input class="form-control" name="sem" type="text"
placeholder="Semester" required="required" pattern="[Sem0-6]{4}" title="Ex. Sem2"/> -->
                    </div>
                    <div class="form-group m-2">
                        <select class="form-control" name="course">
                            <option value="MCA">MCA</option>
                            <option value="MBA">MBA</option>
                        </select>

                    </div>
                    <div class="form-group m-2">
                        <input class="btnbtn-primary px-3" type="submit" value="Add"/>
                    </div>
                </form>
                <%
                try{
                    String driver ="org.postgresql.Driver";

                    String url ="jdbc:postgresql://localhost:5434/postgres";
```

```
                    String username ="postgres";
                    String password ="ravita123";

                Connection con=null;
               Class.forName(driver).newInstance();
                    con = DriverManager.getConnection(url,username,password);

                    if(request.getParameter("rno") != null){
                         PreparedStatementps = con.prepareStatement("insert into
student(rno, name, semester, course) values(?,?,?,?)");
                         ps.setString(1,
request.getParameter("rno").toString().toUpperCase());
                          ps.setString(2, request.getParameter("name").toString());
                          ps.setString(3, request.getParameter("sem").toString());
                          ps.setString(4, request.getParameter("course").toString());
                          if(ps.executeUpdate() > 0){
                          %>
                          <p>Student Added Successfully.</p>
                          <%
                          }else {
                          %>
                          <p>Failed to Add Student.</p>
                          <%
                          }
                    }
               }catch(Exception e){
               System.out.println(e);
               }
               %>
          </center>
</body>
</html>
```

Filename-Update.jsp

```
<%@page import="java.sql.PreparedStatement"%>
<%@page import="java.sql.Connection"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.Statement"%>
<%@ page language="java"contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>

<head>
```

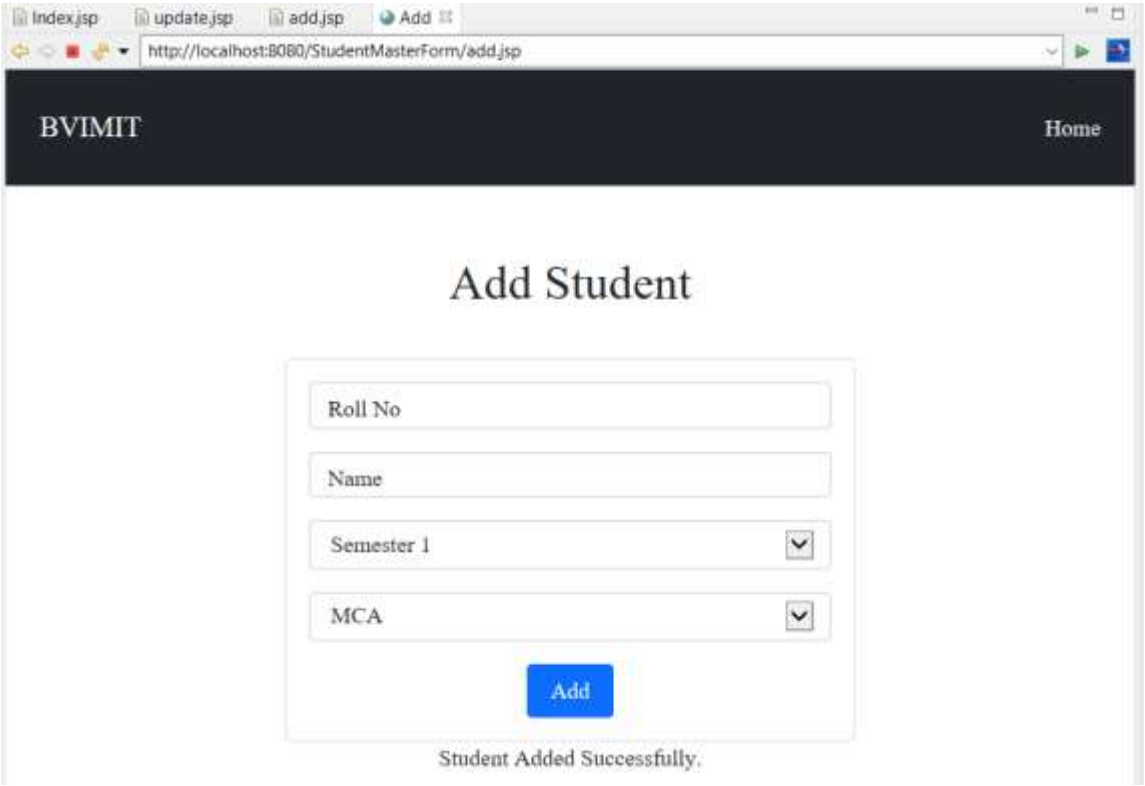
```
 <meta charset="ISO-8859-1">
<title>Update</title>
<!-- CSS only -->
 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.4.0/font/bootstrap-
 icons.css">
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
 beta2/dist/css/bootstrap.min.css"rel="stylesheet" integrity="sha384-
 BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlolflfl"crossorigin="anon
 ymous">
<!-- JavaScript Bundle with Popper -->
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/js/bootstrap.bundle.min.js"
 integrity="sha384-
 b5kHyXgcpbZJO/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S9FOnJ0"crossorigin=
 "anonymous"></script>
</head>
<body>
        <center>
        <nav class="navbar navbar-dark bg-dark p-4">
                <a class="navbar-brand mb-0 h1">BVIMIT</a>
                <ul class="navbar-nav">
                        <li class="nav-item">
                                <a class="navbar-link text-light text-decoration-
 none"href='Index.jsp'>Home</a>
                        </li>
                </ul>
        </nav>
        <br><br>
        <br>

        <form action="update.jsp" method="post" class="card p-2" style="width: 400px">
                <% try{
                                String driver ="org.postgresql.Driver";
                                String url ="jdbc:postgresql://localhost:5434/postgres";
                                String username ="postgres";
                                String password ="ravita123";

                         Connection con=null;
                        Class.forName(driver).newInstance();
                                con = DriverManager.getConnection(url,username,password);
                                System.out.println("Opened database successfully");
                                if(request.getParameter("id")!=null){
                                        Statement stmt = con.createStatement();

                                       ResultSetrs = stmt.executeQuery("SELECT * FROM student
Where id = " + request.getParameter("id"));
                                        if(rs.next()){
```

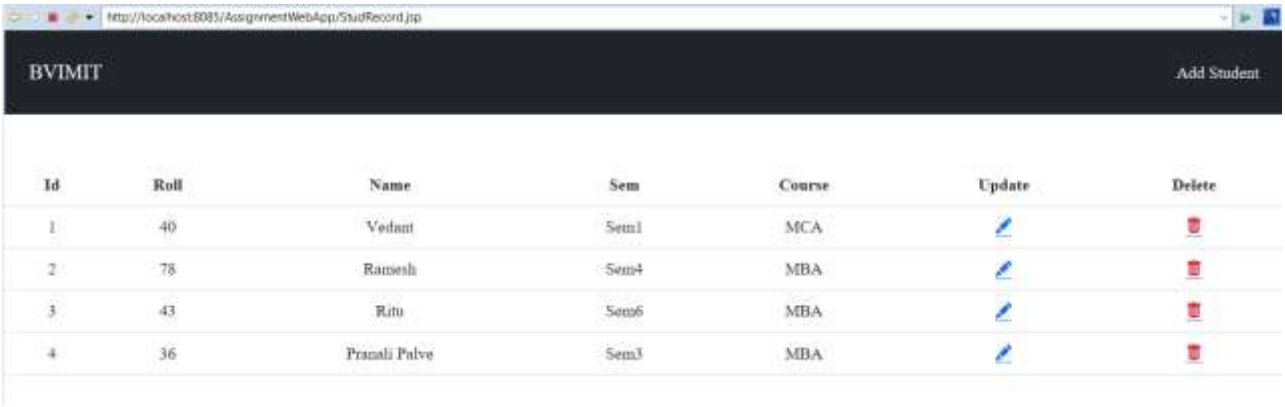
```
                                                %>
                                                <input hidden="hidden" name="uid" type="text"
 value="<%= request.getParameter("id") %>"/>
                                                <div class="form-group m-2">
                                                        <input class="form-control" name="rno"
 type="text" value="<%= rs.getString(2) %>" placeholder="Roll No" required="required" />
                                                </div>
                                                <div class="form-group m-2">
                                                        <input class="form-control" name="name"
 type="text" value="<%= rs.getString(3) %>" placeholder="Name" required="required"  />
                                                </div>
                                                <div class="form-group m-2">
                                <select class="form-control" name="sem" required="required">

                                <option selected disabled="disabled" value="<%= rs.getString(4)%>"><%=
rs.getString(4)%></option>
                                <option value="Sem1">Semester 1</option>
                                <option value="Sem2">Semester 2</option>
                                <option value="Sem3">Semester 3</option>
                                <option value="Sem4">Semester 4</option>
                                <option value="Sem5">Semester 5</option>
                                <option value="Sem6">Semester 6</option>
                        </select>
                                                        <!--<input class="form-control" name="sem"
type="text" placeholder="Semester" required="required" pattern="[Sem0-6]{4}" title="Ex.
Sem2"/> -->
                                                </div>
                                                <div class="form-group m-2">

                                                        <select class="form-control" name="course"
 required="required">
                                                                <option selected disabled="disabled"
 value="<%= rs.getString(5)%>"><%= rs.getString(5)%></option>
                                                                <option value="MCA">MCA</option>
                                                                <option value="MBA">MBA</option>
                                                        </select>
                                                </div>
                                                <div class="form-group m-2">
                                                        <input class="btnbtn-primary px-3"
 type="submit" value="Update"/>
                                                </div>
                                                <%
                                        }else{
                                                %>
                                                <%
                                        }

                                }else
```

```
if(request.getParameter("rno")!=null){

Statement stmt = con.createStatement();
System.out.println(request.getParameter("rno"));
System.out.println(request.getParameter("name"));
System.out.println(request.getParameter("sem"));
System.out.println(request.getParameter("course"));

String query = "";
PreparedStatementps = null;

if(request.getParameter("sem") == null
&&request.getParameter("course") == null){
query = "UPDATE student SET rno= ?, name = ?
WHERE id = '" + request.getParameter("uid") + "'";
ps = con.prepareStatement(query);
ps.setString(1,
request.getParameter("rno").toString().toUpperCase());
ps.setString(2,
request.getParameter("name").toString());
}
else if(request.getParameter("sem") == null){
query = "UPDATE student SET rno= ?, name = ?,
course = ? WHERE id = '" + request.getParameter("uid") + "'";
ps = con.prepareStatement(query);
ps.setString(1,
request.getParameter("rno").toString().toUpperCase());
ps.setString(2,
request.getParameter("name").toString());
ps.setString(4,
request.getParameter("course").toString());
}
else if(request.getParameter("course") == null){
query = "UPDATE student SET rno= ?, name = ?,
semester = ? WHERE id = '" + request.getParameter("uid") + "'";
ps = con.prepareStatement(query);
ps.setString(1,
request.getParameter("rno").toString().toUpperCase());
ps.setString(2,
request.getParameter("name").toString());
ps.setString(3,
request.getParameter("sem").toString());
}else
{
query = "UPDATE student SET rno= ?, name = ?,
semester = ?, course = ? WHERE id = '" + request.getParameter("uid") + "'";

ps = con.prepareStatement(query);
```

```
                                                ps.setString(1,
request.getParameter("rno").toString().toUpperCase());
                                                 ps.setString(2,
request.getParameter("name").toString());
                                                ps.setString(3,
request.getParameter("sem").toString());
                                                ps.setString(4,
request.getParameter("course").toString());
                                                 }

                                        System.out.println(query); int
                                        val = ps.executeUpdate();
                                        System.out.println("val : " + val);
                                        if(val> 0){
                                                 out.write("<script>alert('Updation
Successful.');</script>");
                                                 out.write("<script>window.location.href =
'Index.jsp';</script>");
                                        }else {
                                                 out.write("<script>alert('Updation
Unsuccessful.');</script>");
                                                 out.write("<script>window.location.href =
'Index.jsp';</script>");
                                        }
                                }
                        }catch(Exception e){
                                e.printStackTrace();
                        }
                %>
        </form>
</center></body></html>
```

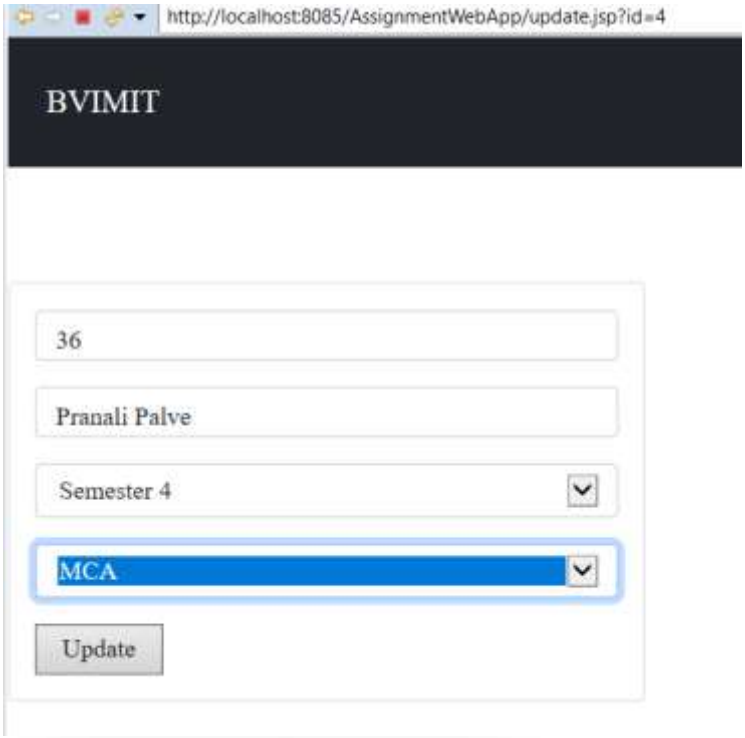
After click on Add

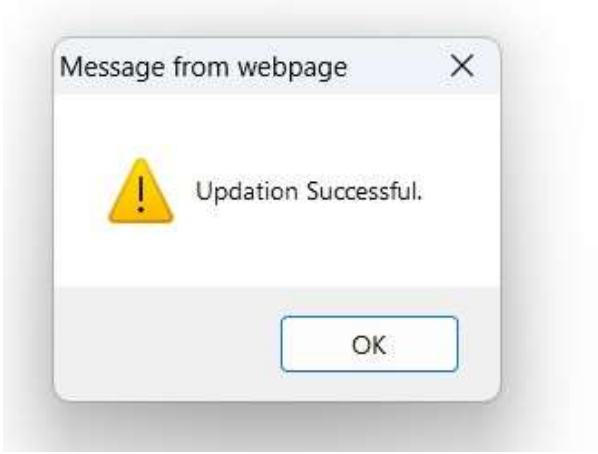




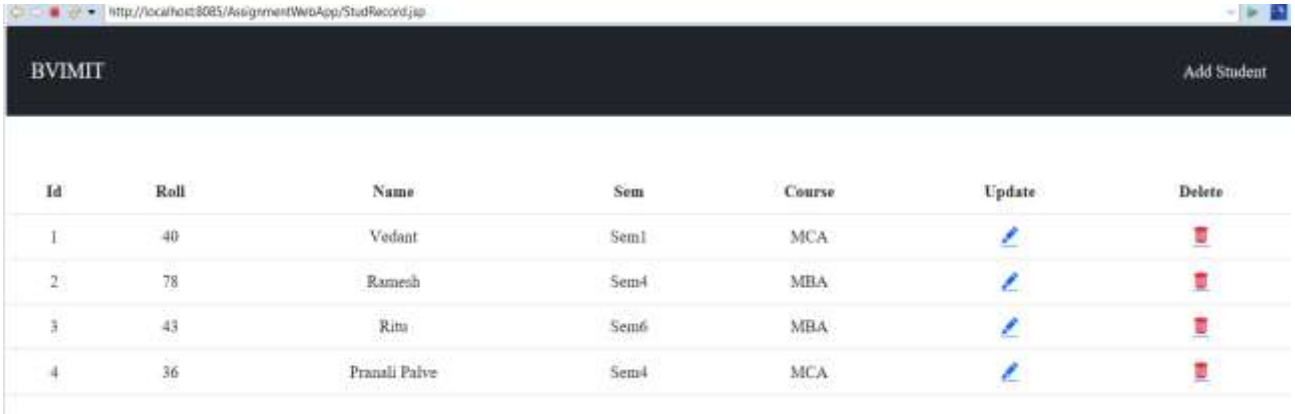
Id	Roll	Name	Sem	Course	Update	Delete
1	40	Vedant	Sem1	MCA		
2	78	Ramesh	Sem4	MBA		
3	43	Ritu	Sem6	MBA		
4	36	Pranali Palve	Sem3	MBA		

Update data






```
Opened database successfully
36
Pranali Palve
Sem4
MCA
UPDATE student SET rno= ?, name = ?, semester = ?, course = ? WHERE id = '4'
val : 1
Opened database successfully
```



Id	Roll	Name	Sem	Course	Update	Delete
1	40	Vedant	Sem1	MCA		
2	78	Ramesh	Sem4	MBA		
3	43	Ritu	Sem6	MBA		
4	36	Pranali Palve	Sem4	MCA		

Database:

CREATE :
create table student(
id SERIAL PRIMARY KEY,
rnovarchar(4),
name varchar(20),
semester varchar(10),
course varchar(5)
)

Query Query History

```
create table student
(
id SERIAL PRIMARY KEY,
rno varchar(4),
name varchar(20),
semester varchar(10),
course varchar(5)
);
```

Data Output Messages Notifications

```
CREATE TABLE

Query returned successfully in 74 msec.
```

d. Design loan calculator using JSP which accepts Period of Time (in years) and Principal Loan Amount. Display the payment amount for each loan and then list theloan balance and interest paid for each payment over the term of the loan for the following time period and interest rate:

1. **1 to 7 year at 5.35%**
2. **8 to 15 year at 5.5%**
3. **16 to 30 year at 5.75%**

Input:

Cal.jsp

```
<%@ page language="java"contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html><head>
 <meta charset="ISO-8859-1"><title>Load Calculator</title>
<!-- CSS only -->
 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.4.0/font/bootstrap-
 icons.css">
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
 beta2/dist/css/bootstrap.min.css"rel="stylesheet"integrity="sha384-
 BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlolflfl"crossorigin="anon
 ymous">
<!-- JavaScript Bundle with Popper -->
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
 beta2/dist/js/bootstrap.bundle.min.js"integrity="sha384-
 b5kHyXgcpbZJO/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S9FOnJ0"crossorigin=
 "anonymous"></script>
</head>
<body><br>
<h1><center>Loan Calculator</center></h1><br>
 <form name="loancal" action="Test.jsp" method="post" class="card p-2 m-auto" style="width:
 400px;">

 <div class="form-group m-2">
        Principal Loan Amount:
        <input class="form-control" type="text" name="pamt" placeholder="Enter Principal
 Amount">
</div>
 <div class="form-group m-2">
        Tenure (in years):
        <input class="form-control" type="text" name="time" placeholder="Enter period of time">
</div>
 <input class="form-control p-2" type="submit" value="Calculate">
```

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


    Text.jsp
 <%@ page language="java"contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
 <meta charset="ISO-8859-1">
<title>Calculated Loan</title>
<!-- CSS only -->
 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.4.0/font/bootstrap-
 icons.css">
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
 beta2/dist/css/bootstrap.min.css"rel="stylesheet"integrity="sha384-
 BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlolflfl"crossorigin="anon
 ymous">
<!-- JavaScript Bundle with Popper -->
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
 beta2/dist/js/bootstrap.bundle.min.js"integrity="sha384-
 b5kHyXgcpbZJO/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S9FOnJ0"crossorigin=
 "anonymous"></script>
</head>
<body>

<%
  String p_amt=request.getParameter("pamt");
  String  tenure=request.getParameter("time");
  float pr_amt=Float.parseFloat(p_amt);
  float period=Float.parseFloat(tenure);
  double loan_balance, interest, emi;

 out.println("<br><br><div class='card p-3 m-auto' style='width:400px;'><center><h1>Loan
Details</h1><hr>");

  if(period>=1 && period<=7)
  {
          emi=pr_amt*0.0535;
          interest=pr_amt*0.0535*period;
          loan_balance=pr_amt+interest;
          out.println("EMI : " + emi + " Rs.");
          out.println("<br>Total Interest : " + interest + " Rs.");
          out.println("<br>Loan Balance : " + loan_balance + " Rs.");
  }
  if(period>=8 && period<=15)
  {
          emi=pr_amt*0.055;
```

```
            interest=pr_amt*0.0535*period;
            loan_balance=pr_amt+interest;

            out.println("EMI : " + emi + " Rs.");
            out.println("<br>Total Interest : " + interest + " Rs.");
            out.println("<br>Loan Balance : " + loan_balance + " Rs.");
  }
  if(period>=16 && period<=30){
            emi=pr_amt*0.0575;
            interest=pr_amt*0.0535*period;
            loan_balance=pr_amt+interest;
            out.println("EMI : " + emi + " Rs.");
            out.println("<br>Total Interest : " + interest + " Rs.");
            out.println("<br>Loan Balance : " + loan_balance + " Rs.");
  }
  out.println("</center></div>");
%>
</body>
</html>
```

OUTPUT:

http://localhost:8085/AssignmentWebApp/Text.jsp

Loan Details

EMI : 1337.5 Rs.

Total Interest : 4012.5 Rs.

Loan Balance : 29012.5 Rs.

e. Application form for change of Study Center which can be filled by any student who wants to change his/ her study center. Make necessary assumptions.

Input:

Study_center.jsp

```
<%@ page language="java"contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
 <meta charset="ISO-8859-1">
<title>Study Center</title>
</head>
<body>
<h1><center>Study Center</center></h1>
<hr>

 <form action="register.jsp" method="post">
<fieldset>
<legend>Personal Details</legend>
<table style="with: 50%">
<tr>
<td>Register No:</td>
 <td><input type="number" /></td>
</tr>
<tr>
<td>Name:</td>
 <td><input type="text" placeholder="first Name" /></td>
<td>Middle:<td>
 <td><input type="text" placeholder="middle Name" /></td>
<td>Last:</td>
 <td><input type="text" placeholder="last Name" /></td>
</tr>
<tr>
<td>Address:</td>
 <td><input type="text" /></td>
</tr>
<tr>
<td>Email-id:</td>
 <td><input type="text" placeholder="ravitapatil919@gmail.com" /><br><br></td>
</tr>

<tr>

<td>Password</td>
 <td><input type="password" name="password" /></td>
```

```
</tr>

<tr>
<td>confirm Password</td>
  <td><input type="password" name="cpassword" /></td>
</tr>
<tr>
<tr>
<td>Contact No:</td>
  <td><input type="text" /></td>
</tr>
<tr>
<td>DOB:</td>
  <td><input type="text" placeholder="4/11/1999" /><br><br></td>
</tr>
<tr>
<td>Gender:</td>
<td>Male:</td>
  <td><input type="radio"  Name="Gender" /></td>
<td>Female:</td>
  <td><input type="radio"  Name="Gender" /></td>
</tr>
<tr>
<td>City</td>
  <td><select name="City">
<option value="-1" selected>select..</option>
  <option value="New Delhi">PANVEL</option>
  <option value="Mumbai">KAMOTHE</option>
<option value="Goa">NERUL</option>
<option value="Patna">VASHI</option>
</select></td>
</tr>
</table>
</fieldset>
<fieldset>
<legend>Educational details</legend>
<table style="with: 50%">
<tr>
<td>Qualification</td>
</tr>
<tr>
<td>SSC:</td>
  <td><input type="checkbox"></td>
<td>HSC:</td>
  <td><input type="checkbox"></td>

<td>Post Graduation:</td>
  <td><input type="checkbox"></td>
```

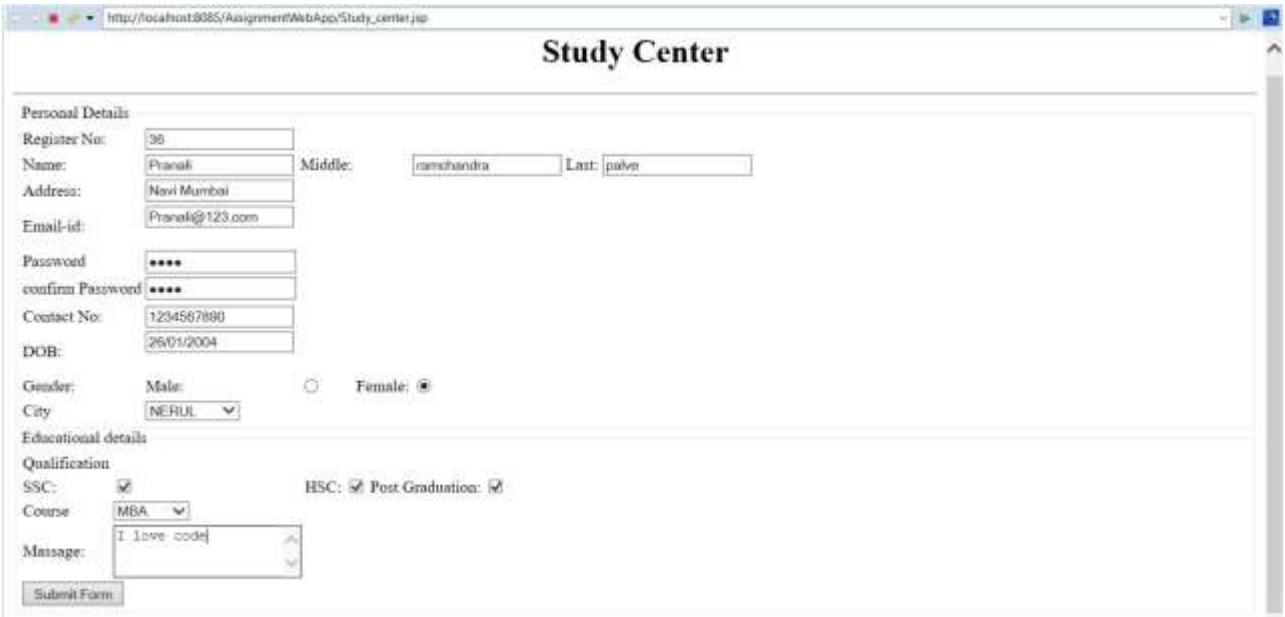
```
</tr>
<tr>
<td>Course</td>
  <td><select name="Course">
<option value="-1" selected>select..</option>
  <option value="B.Tech">B.TECH</option>
<option value="MCA">MCA</option>
<option value="MBA">MBA</option>
<option value="BCA">BCA</option>
</select></td>
</tr>
<tr>
<td>Massage:</td>
<td><textarea rows = "3"></textarea></td>
</tr>
<tr>
  <td colspan="2"><input type="submit" value="Submit Form" /></td></tr>
</table>
</fieldset><br>

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

Filename-Register.jsp

```
  <%@ page language="java"contentType="text/html; charset=ISO-8859-1"
      pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
  <meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<%
String n=request.getParameter("first name");
String str1=request.getParameter("password");
String str2=request.getParameter("cpassword");
if(str1.equals(str2))
   {
out.println("Your request to change Study Center from has been sent to the Administrator.");

   }

   else
   {
```

```
  out.println("<h3>Sorry, your password is mismatched</h3>");
  }
%>
</body>
</html>
```

OUTPUT-



After click on Submit Form

Your request to change Study Center from has been sent to the Administrator.

If password is wrong

Sorry, your password is mismatched

f. Write a JSP program that demonstrates the use of JSP declaration, scriptlet, directives, expression, header and footer.

Input:

main.jsp

```
<%@ page language="java"contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
 <meta charset="ISO-8859-1">
<title>JSP EXAMPLE</title>
</head>
<body>
 <%@ include file = "header.jsp" %>
<center>

<%! int data=50; %>
<%= "Value of the variable is:"+data %>

<%!
double circle(int n){ return 3.14*n*n;}
%></br>
<%= "Area of circle is:"+ circle(3) %></br>

<%!
int rectangle(int l,int b){ return l*b;}
%>
<%= "Area of rectangle is:"+rectangle(3,4
) %></br>

<%!
 int perimeter(int x,int y){
  int peri=2*(x+y);
return peri;}
%>
<%= "Perimeter of rectanlge:"+perimeter(5,6
) %></br>
  <p>Thanks for visiting my page.</p>
</center>
 <%@ include file = "footer.jsp" %>

</body>
        </html>
```

header.jsp

```
 <%@ page language="java"contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<%!
  int pageCount = 0;
  void addCount() {
  pageCount++;
  }
%>
<% addCount(); %>

<html>
<head>
 <meta charset="ISO-8859-1">
<title>JSP declaration, scriptlet, directives, expression, header and footer Example</title>
</head>
<body>
  <center>
  <h2>The include Directive Example</h2>
  <p>This site has been visited <%= pageCount %>times.</p>
  </center>
  <br/><br/>

</body>
</html>
```

footer.jsp

```
 <%@ page language="java"contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
 <meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<br/><br/>
  <center>
  <p>Copyright 2021</p>
  </center>

</body>
</html>
```

OUTPUT-


http://localhost:8085/AssignmentWebApp/main.jsp
The include Directive Example
This site has been visited 1times.
Value of the variable is:50
Area of circle is:28.259999999999998
Area of rectangle is:12
Perimeter of rectanlge:22
Thanks for visiting my page.
Copyright 2021

g. Write a JSP program that demonstrates the use of cookies.

Input:

Test2.jsp

```
<%@page language="java" contentType="text/html;charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<!DOCTYPEhtml>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="SessionCookie.jsp" method="post">
<table>
<tr>
<td>UserName:</td>
<td><input type="text" name="username"></td>
</tr>
<tr>
<td>Email:</td>
<td><input type="text" name="email"/></td></tr>
<tr>
<td colspan="2"><input type="submit" value="SubmitForm"/></td>
</tr>
</table>
</form>
</body>
</html>
```

SessionCookie.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

<%
if(request.getParameter("username")!=null){
```

```
Cookie username = new Cookie("username", request.getParameter("username"));
Cookie email=new Cookie("email",request.getParameter("email"));

session.setAttribute("username", request.getParameter("username"));
session.setAttribute("email",request.getParameter("email"));


//Add both the cookies in the response header.
response.addCookie( username );response.addCookie(email);

Cookie cookie = null;
Cookie[] cookies= null;

//Get an array of Cookies associated with the this domain
cookies=request.getCookies();

  if(cookies!=null) {
            out.println("<h2>Retrived From Cookie</h2>");


for(int i=1;i<cookies.length;i++){

out.print(cookies[i].getValue()+"");
}
   }else
{
out.println("<h2>No cookies founds</h2>");
}
}
%>
</body>
</html>
```

Set Cookie Example

Enter your name:

Set Cookie

Cookie has been set successfully. Go to Display Cookie

Set Cookie Example

Enter your name: pranali

Set Cookie

Display Cookie Example

Username Cookie: pranali

Set New Cookie

Assignment No. 7

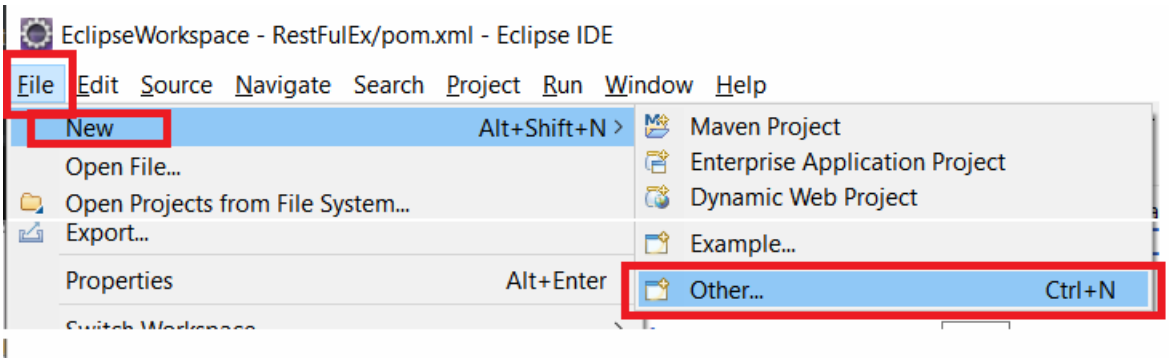
Spring Framework

1. Write a program to print “Hello World” using spring framework.
2. Write a program to demonstrate dependency injection via setter method.
3. Write a program to demonstrate dependency injection via Constructor.
4. Write a program to demonstrate Autowiring

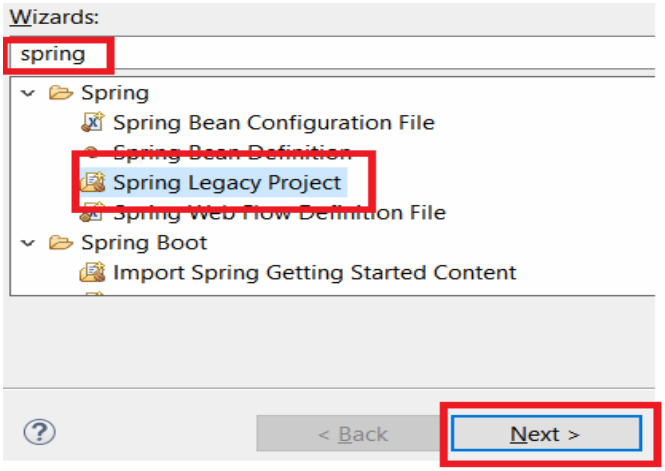
Steps to Create Spring Legacy Project

Step 1 : Creating Spring Legacy Project.

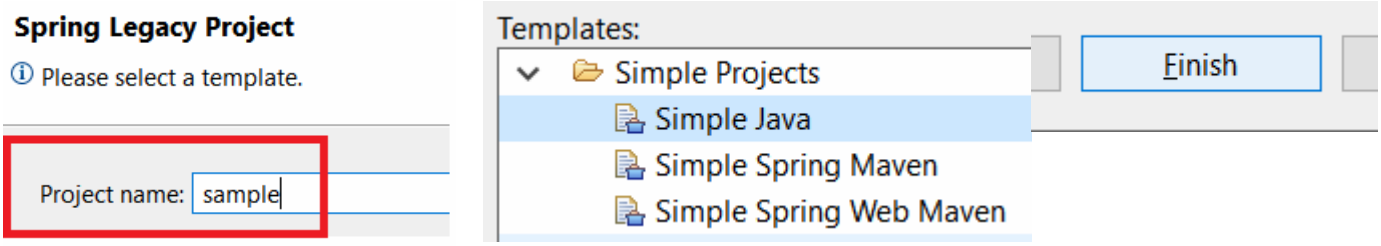
1.1 : Open Eclipse. Go To File > New > Other.



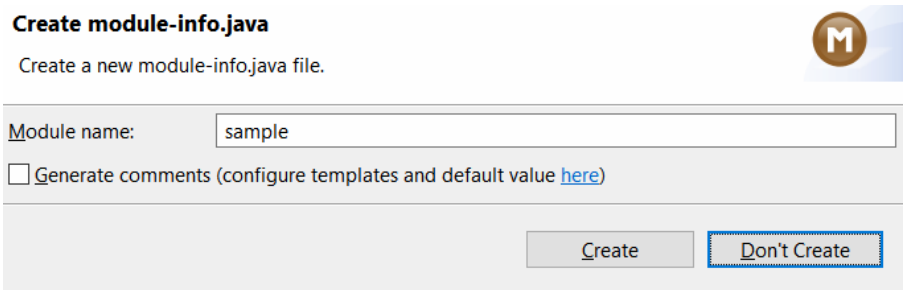
1.2 : Search for 'spring' and Select 'Spring Legacy Project'. Then Click on Next.



1.3 : ChooseProject Name of your wish, below there select **Simple Java** & simply Finish.

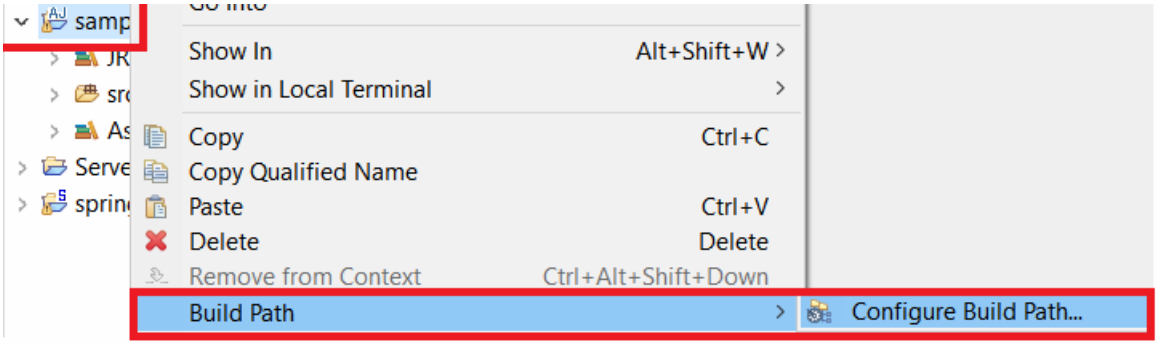


1.4 : If asked for Creating module-info.java file, click on **Don't Create**.

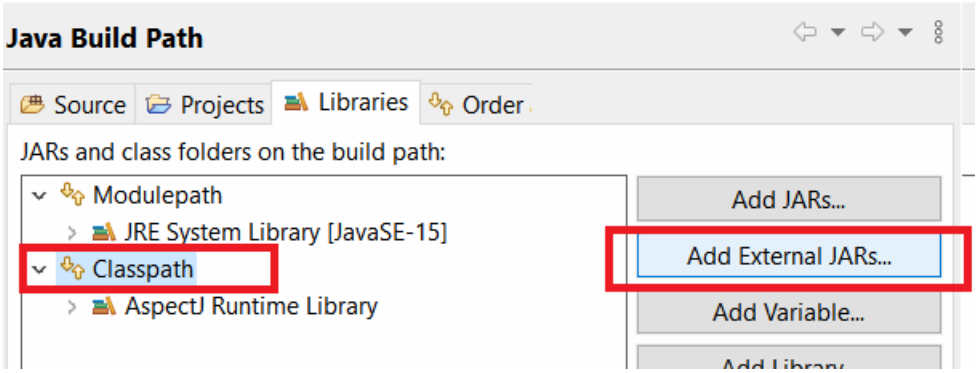


Step 2 : Adding the Spring Libraries.

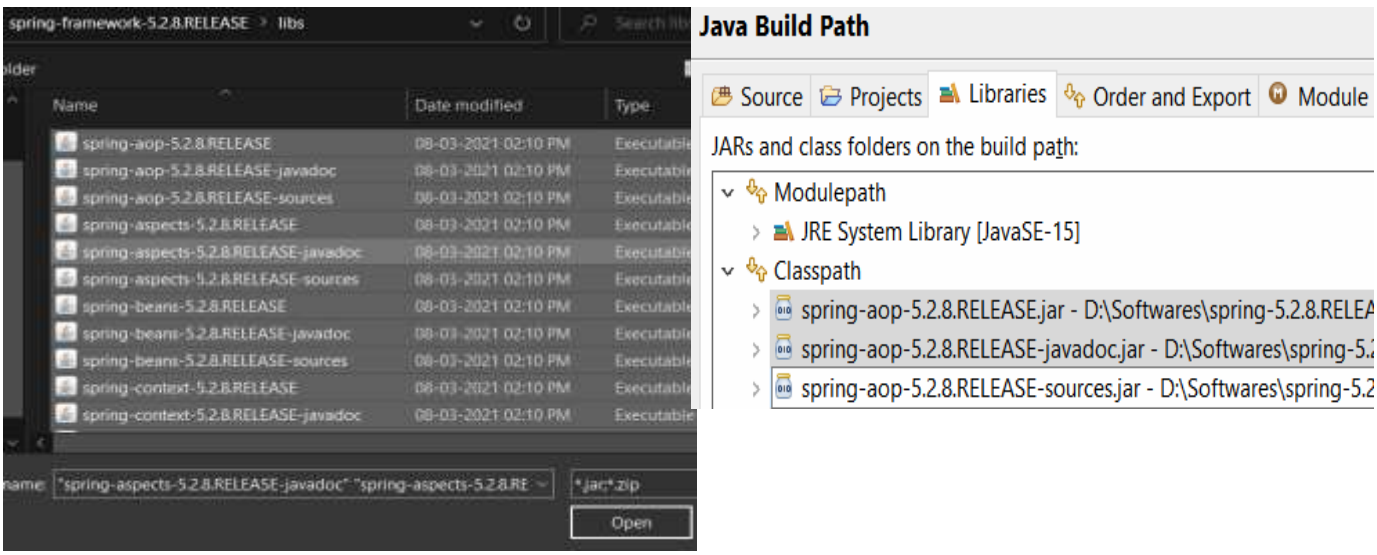
2.1 : Right click on your Newly created Spring Legacy project, Choose Build Path > Configure Build Path.



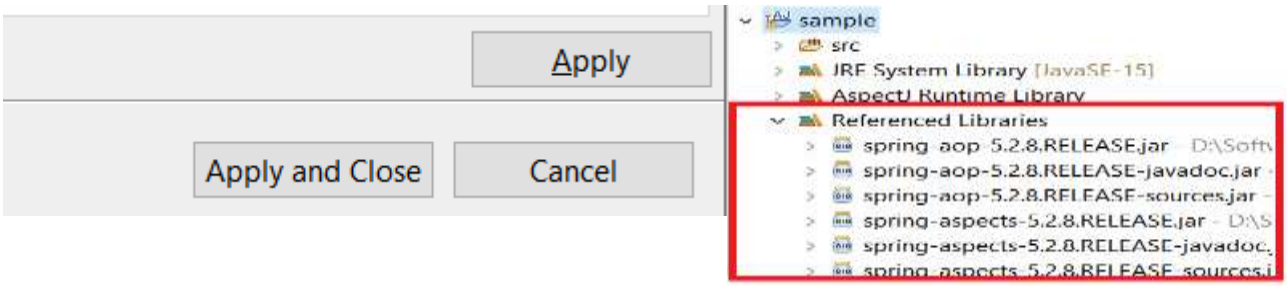
2.2 On Java Build Path wizard, Choose **Classpath** and then select **Add External JARs.**



2.3 : Choose all the Spring Libraries you've downloaded, and click on OPEN. This will add all libraries to Classpath.



2.4 Finally click on Apply & Close, now you are ready to work with Spring Legacy Project.


Apply
Apply and Close
Cancel
sample
src
JRE System Library [JavaSE-15]
AspectJ Runtime Library
Referenced Libraries
spring-aop-5.2.8.RELEASE.jar
spring-aop-5.2.8.RELEASE-javadoc.jar
spring-aop-5.2.8.RELEASE-sources.jar
spring-aspects-5.2.8.RELEASE.jar
spring-aspects-5.2.8.RELEASE-javadoc.
spring-aspects-5.2.8.RELEASE-sources.j

PROBLEM STATEMENT 1 : Write a program to print “Hello World” using spring framework.

Input:

HelloWorld.java

```
package spring1;

public class HelloWorld

{String name;

public String getName()

{return name;

}

public void setName(String name)

{this.name = name;

}

@Override

public String toString() {

return "Hello World, I'm " + name + ".";

}

}
```

appctx3.xml

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

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="hw" class="spring1.HelloWorld">

<property name="name" value="Pranali B36"/>

</bean>
```

```
</beans>
```

TestHelloWorld.java

```
package spring1;

import org.springframework.context.support.ClassPathXmlApplicationContext; public class
TestHelloWorld {

public static void main(String[] args) {

ClassPathXmlApplicationContext app = new

ClassPathXmlApplicationContext("appctx3.xml");

HelloWorld hw = (HelloWorld) app.getBean("hw");

System.out.println(hw.toString());

}

}
```

Output :

```
<terminated> TestHelloWorld [Java Application] C:\Program Files\Java\jre1.8.0_202\bin\javaw.exe (29-Nov-2024, 12:29:29 am)
Hello World, I'm Pranali B36.
```

PROBLEM STATEMENT 2 : Write a program to demonstrate dependency injection via setter method.

Input:

Singer.java

```
package spring1;

public class Singer {

String name;

int age;

public String getName()

{return name;



}

public void setName(String name)

{this.name = name;

}

public int getAge()

{return age;

}

public void setAge(int age)

{this.age = age;

}

void displayInfo()

{

System.out.println("Name:" +name+" Age:" +age);

}

}
```

appctx.xml

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

<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd"> <bean id="Singer"
class="spring1.Singer">

<property name="name" value="Pranali B36"></property>

<property name="age" value="20"></property>

</bean>

</beans>
```

SingerTest.java

```
package spring1;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext; public
classSingerTest {

private static ApplicationContext ctx;

public static void main(String[] args) {

// TODO Auto-generated method stub

ctx=new

ClassPathXmlApplicationContext("appctx.xml");Singer

singer=(Singer)ctx.getBean("Singer");

singer.displayInfo();

}

}
```

Output :

```
<terminated> SingerTest [Java Application] C:\Program Files\Java\jre1.8.0_202\bin\javaw.exe (29-Nov-2024, 12:38:59 am)
Name:Pranali B36 Age:20
```

PROBLEM STATEMENT 3 : Write a program to demonstrate dependency injection via Constructor.

Input:

Address.java

```
package depinjectionbycons;


public class Address {


private String city;

private String state;

private String

country;

public Address(String city, String state, String country) {
        super();
        this.city = city;
        this.state = state;
        this.country =
        country;

}

public String toString(){

        return city+" "+state+" "+country;

}

}
```

Employee.java

```
package depinjectionbycons;

public class Employee {

private int id;

private String name;

private Address address;

public Employee() {System.out.println("def cons");}

public Employee(int id, String name, Address address) {

super();

this.id = id;

this.name = name;

this.address = address;

}

void show(){

System.out.println(id+" "+name);

System.out.println(address.toString());
}
}
```

applicationContext.xml

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

<beans

xmlns="http://www.springframework.org/schema/beans"

 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:p="http://www.springframework.org/schema/p"

 xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">

<bean id="a1" class="depinjectionbycons.Address">
```

```
<constructor-arg value="Navi Mumbai"></constructor-arg>

<constructor-arg value="Maharashtra"></constructor-arg>

<constructor-arg value="India"></constructor-arg>

</bean>

<bean id="e" class="depinjectionbycons.Employee">

<constructor-arg value="36" type="int"></constructor-arg>

<constructor-arg value="Pranali"></constructor-arg>

<constructor-arg>

<ref bean="a1"/>

</constructor-arg>

</bean>

</beans>
```

Test.java

```
package depinjectionbycons;
import org.springframework.beans.factory.BeanFactory;

import org.springframework.beans.factory.xml.XmlBeanFactory;
importorg.springframework.core.io.ClassPathResource;

import org.springframework.core.io.Resource;

public class Test {

public static void main(String[] args) {

Resource r=new ClassPathResource("applicationContext.xml");
@SuppressWarnings("deprecation")

BeanFactory factory=new

XmlBeanFactory(r); Employee

s=(Employee)factory.getBean("e");s.show();

}

}
```

Output:

```
<terminated> ETest [Java Application] C:\Program Files\Java\jre1.8.0_202\bi
36 Pranali
Navi Mumbai Maharashtra India
```

PROBLEM STATEMENT 4 :Write a program to demonstrate Auto-wiring.

Input:

B.java

```
package org.bvimitt;
public class B {
        B(){System.out.println("b is created");}
        void print(){System.out.println("hello
        b");}
}
```

A.java

```
package org.bvimit;
public class A {
        B b;
        A(){System.out.println("a is
        created");}public B getB() {
            return b;
        }
        public void setB(B b)
            {this.b = b;
        }
        void print(){System.out.println("hello
        a");}void display(){
            print();
            b.print();
        }
}
```

applicationContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans
    xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:p="http://www.springframework.org/schema/p"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">
<bean id="b" class="org.bvimit.B"></bean>
<bean id="a" class="org.bvimit.A" autowire="byName"></bean>
</beans>
```

Test.java

```
package org.bvimit;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Test {
public static void main(String[] args) {
    ApplicationContext context=new ClassPathXmlApplicationContext("applicationContext.xml
```

```
  ");

    A a=context.getBean("a",A.class);
a.display();

  }
}
```

OUTPUT :

```
<terminated> Test [Java Application] C:\Program Files\Java\jre1.8.0_202\bin\javaw.exe (29-Nov-2024, 1:07:10 am)
Pranali B36
b is created
A is Created
hello A
Hello B
```

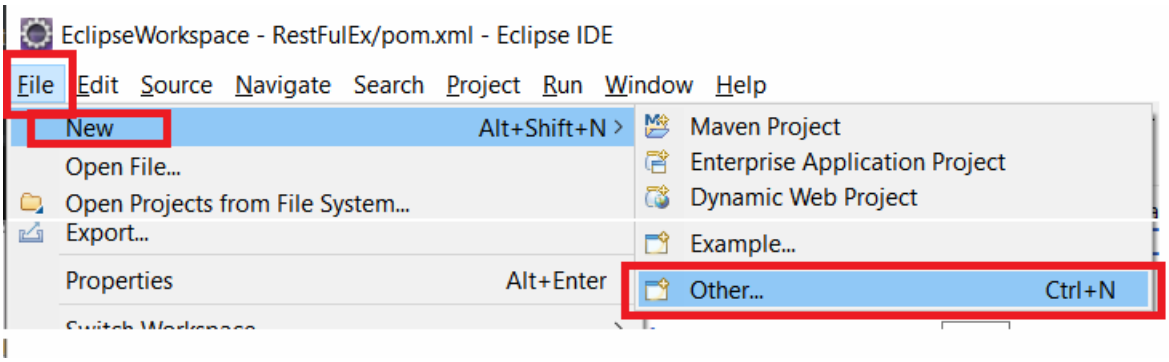
Assignment No 8

1. **Write a program to demonstrate Spring AOP – before advice.**
2. **Write a program to demonstrate Spring AOP – after advice.**
3. **Write a program to demonstrate Spring AOP – around advice.**
4. **Write a program to demonstrate Spring AOP – after returning advice.**
5. **Write a program to demonstrate Spring AOP – after throwing advice.**
6. **Write a program to demonstrate Spring AOP – pointcuts.**

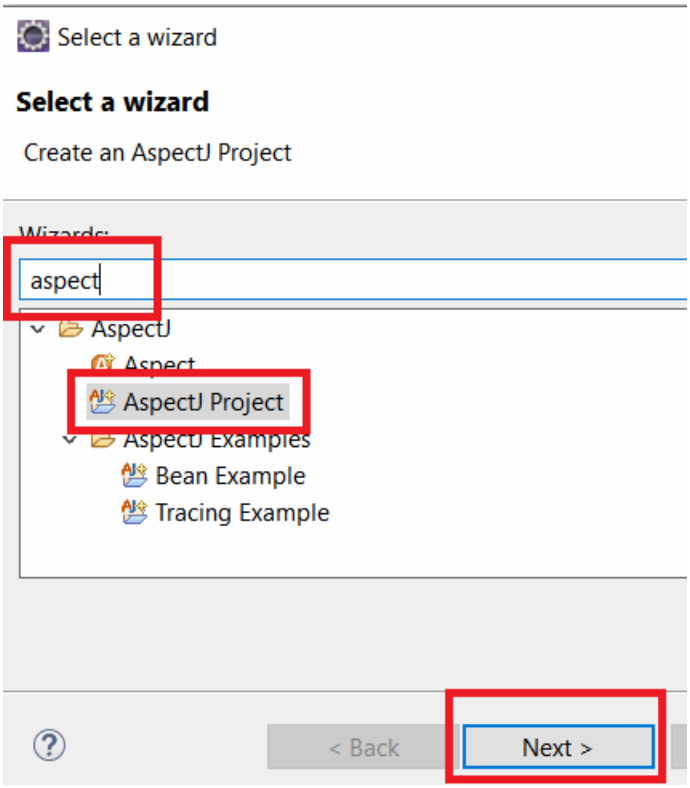
Steps to Create an AOP Project

Step 1 : Creating AspectJ Project.

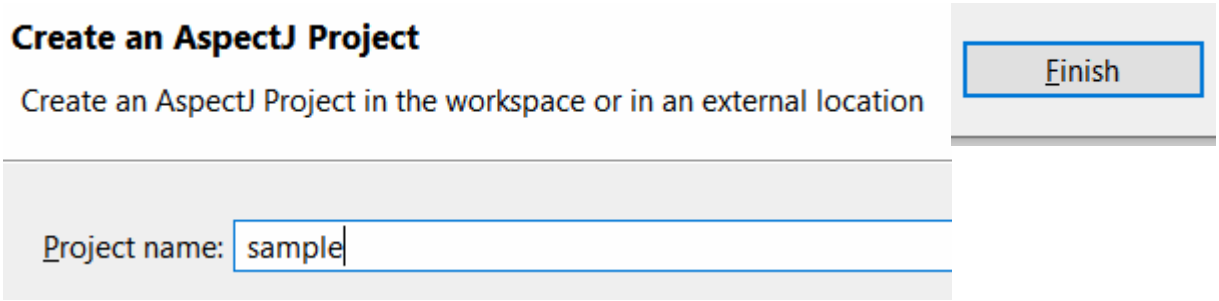
1.1 : Open Eclipse. Go To File > New > Other.



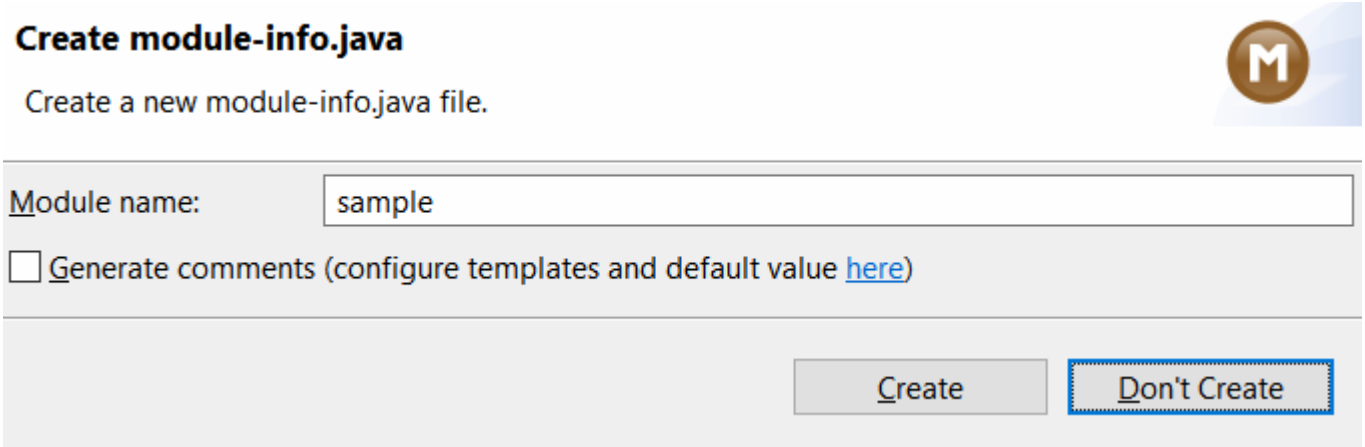
1.2 : Search for 'aspect' and Select 'AspectJ Project'. Then Click on Next.



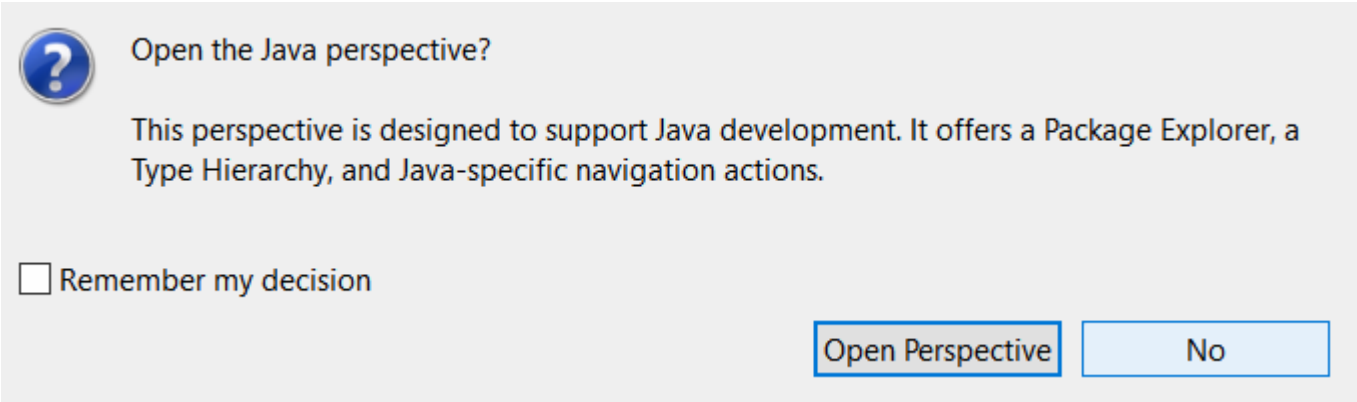
1.3 : Enter Project Name of your wish, and click on Finish.



1.4 : If asked to create module-info.java file, select 'Don't Create'.



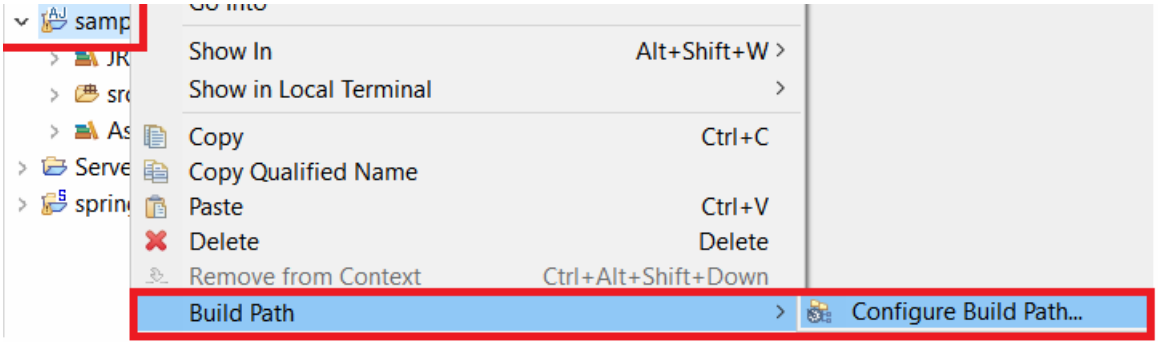
1.5 : Finally if you are asked to Open Java Perspective, just choose **NO.**



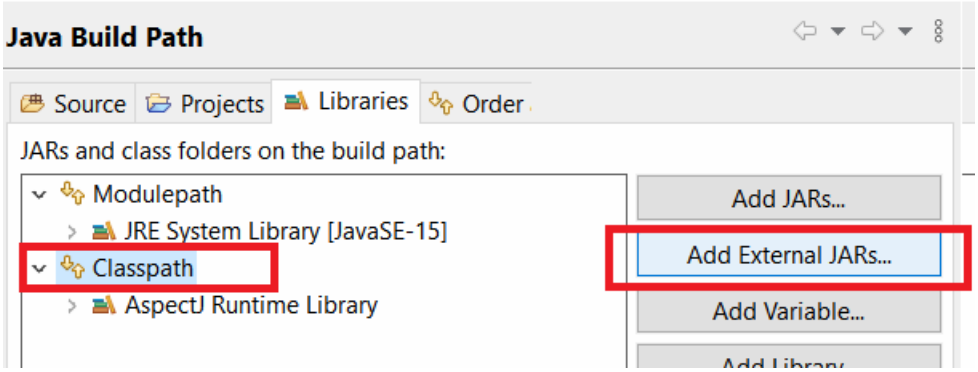
This creates your AspectJ project.

Step 2 : Adding the Spring Libraries.

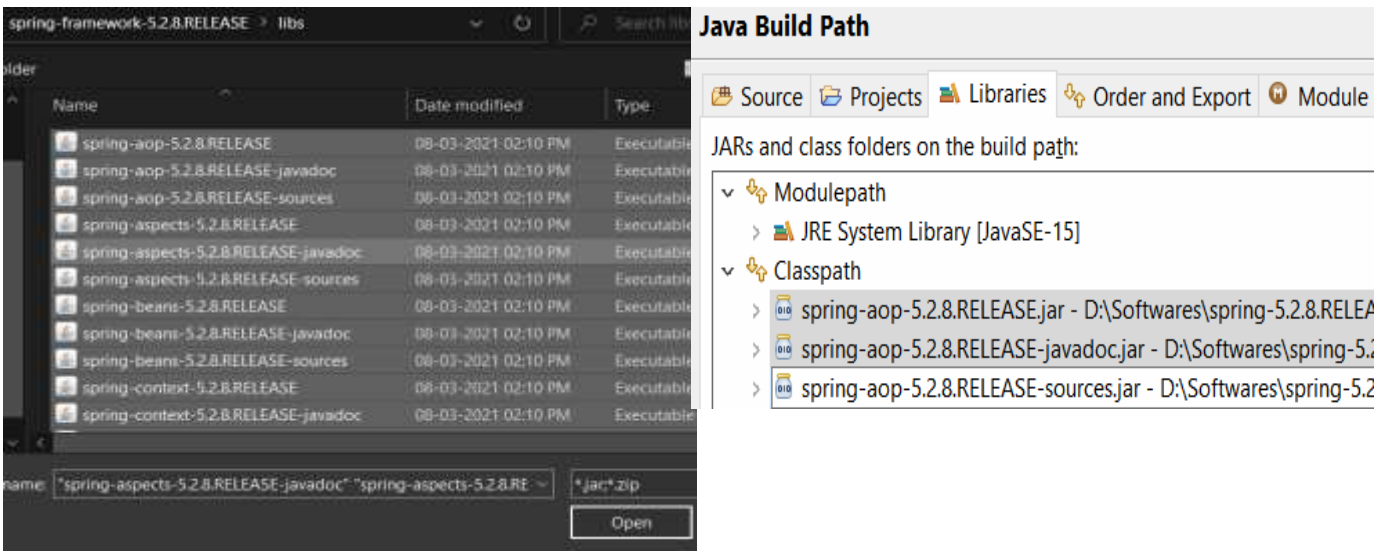
2.1 : Right click on your Newly created AspectJ project, Choose Build Path > Configure Build Path.



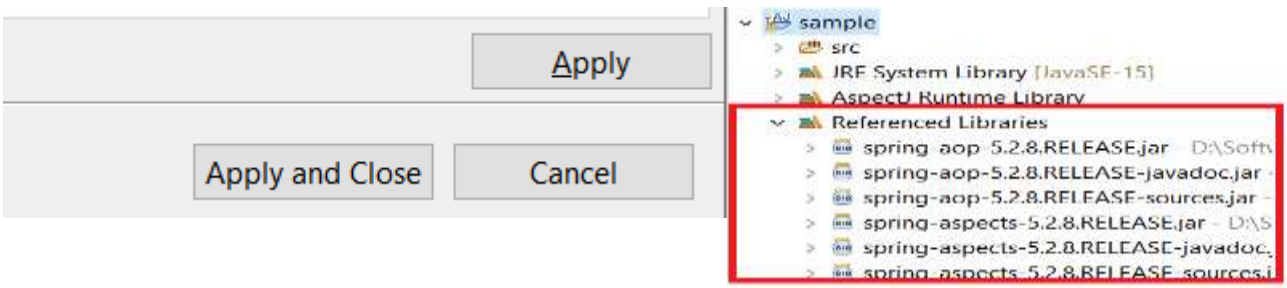
2.2 On Java Build Path wizard, Choose **Classpath**and then select **Add External JARs.**



2.3 : Choose all the Spring Libraries you've downloaded, and click on OPEN. This will add all libraries to Classpath.



2.4 Finally click on Apply & Close, now you are ready to work with Aspects in Spring.



Problem Statement 1 : Write a program to demonstrate Spring AOP – before advice.
Solution : beforeaop.java

```
package Sample;
import org.aspectj.lang.JoinPoint;
import org.aspectj.lang.annotation.Aspect; import org.aspectj.lang.annotation.Before; import
org.aspectj.lang.annotation.Pointcut; @Aspect
public class beforeaop
{


@Pointcut("execution(int beforeoperation.*(..))") public void p(){}
@Before("p()")
public void myadvice(JoinPoint jp)
{
System.out.println("before advice");
}
}
```

beforeoperation.java

```
package Sample;
public class beforeoperation
{
public void msg()
{
System.out.println("method 1");
}
public int m()
{
System.out.println("method 2 with return"); return 2;
}
public int k()
{
System.out.println("method 3 with return"); return 3;
}
}
```

aopctx1.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
<bean id="opBean" class="Sample.beforeoperation"> </bean>
<bean id="trackMyBean" class="Sample.beforeaop"></bean>
<bean class="org.springframework.aop.aspectj.annotation.AnnotationAwareAspectJAutoProxyCreator
">
</bean>
</beans>
```

beforetest.java

```
package Sample;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext; public class beforetest
{
public static void main(String[] args) {

System.out.println("Pranali B36"); ApplicationContext context = new

ClassPathXmlApplicationContext("aopctx1.xml"); beforeoperation e = (beforeoperation)
context.getBean("opBean"); System.out.println("calling m1.          ");
e.msg();
System.out.println("calling m2.        ");
e.m();
System.out.println("calling m3.        ");
e.k();
}
}
```

Output:

```
<terminated> Beforetest (1) [AspectJ/Java Application] C:\java\New folder\bin\javaw.exe (28-Nov-2024, 2:03:40
Pranali B36
calling m1......
method 1
calling m2......
method 2 with return
calling m3......
method 3 with return
```

Problem Statement 2 : Write a program to demonstrate Spring AOP – after advice.
Solution : Afteraopdata.java

```
package Sample;
import org.aspectj.lang.JoinPoint; import org.aspectj.lang.annotation.After;
import org.aspectj.lang.annotation.Aspect; import org.aspectj.lang.annotation.Pointcut; @Aspect
public class Afteraopdata
{
@Pointcut("execution(int afteroperation.*(..))") public void p(){}
@After("p()")
public void myadvice(JoinPoint jp)
{
System.out.println("after advice");
}
}
```

afteroperation.java

```
package Sample;
public class afteroperation
{
public void msg()
{
System.out.println("method 1");
}
public int m()
{
System.out.println("method 2 with return"); return 2;
} public int k()
{
System.out.println("method 3 with return"); return 3;
}
}
```

aopctx.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
<bean id="opBean" class="Sample.afteroperation"> </bean>
```

```
<bean id="trackMyBean" class="Sample.Afteraopdata"></bean>

<bean class="org.springframework.aop.aspectj.annotation.AnnotationAwareAspectJAutoProxyCreator
"></bean>
</beans>
```

aftertest.java

```
package Sample;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext; public class aftertest
{
public static void main(String[] args)
{
System.out.println("Pranali B36");
ApplicationContext context = new ClassPathXmlApplicationContext("aopctx.xml"); afteroperation e =
(afteroperation) context.getBean("opBean"); System.out.println("calling m1.        ");
e.msg();
System.out.println("calling m2.        ");
e.m();
System.out.println("calling m3.        ");
e.k();
}
}
```

Output:

```
<terminated> Aftertest (6) [AspectJ/Java Application] C:\java\New folder\bin\java
Pranali B36
calling m1......
method 1
calling m2......
method 2 with return
calling m3......
method 3 with return
```

Problem Statement 3 : Write a program to demonstrate Spring AOP – around advice.
Solution : Bankaopdata.java

```
package Sample;
import org.aspectj.lang.ProceedingJoinPoint; import org.aspectj.lang.annotation.Around; import
org.aspectj.lang.annotation.Aspect; import org.aspectj.lang.annotation.Pointcut; @Aspect
public class Bankaopdata
{
@Pointcut("execution(* Bank.*(..))") public void a() {}
@Around("a()")
public Object myadvice(ProceedingJoinPoint p)throws Throwable
{
System.out.println("Around concern Before calling actual method"); Object obj=p.proceed();
System.out.println("Around Concern After calling actual method"); return obj;
}
}
```

Bank.java

```
package Sample; public class Bank
{
public void welcome()
{
System.out.println("welcome to bank");
}
public int icici()
{
System.out.println("icici bank interest rate"); return 7;
}
public int pnb()
{
System.out.println("pnb bank interest rate"); return 6;
}
}
```

Bankaopdata.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
<bean id="opBean" class="Sample.Bank"> </bean>
<bean id="trackMyBean" class="Sample.Bankaopdata"></bean>
<bean class="org.springframework.aop.aspectj.annotation.AnnotationAwareAspectJAutoProxyCreator
"></bean>
</beans>
```

Banktest.java

```
package Sample;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext; public class Banktest
{
private static ApplicationContext context; public static void main(String[] args)
{
System.out.println("Pranali B36");
context = new ClassPathXmlApplicationContext("Bankaopdata.xml"); Bank e =(Bank)
context.getBean("opBean"); System.out.println("Calling welcome method...");
e.welcome();
System.out.println("Calling icici method..."); e.icici();
System.out.println("Calling pnb method..."); e.pnb();
}
}
```

Output:

```
<terminated> Banktest (8) [AspectJ/Java Application] C:\java\New folder\bin\javaw.exe (28-Nov-2(
Pranali B36
Calling welcome method...
Around concern Before calling actual method
welcome to bank
Around Concern After calling actual method
Calling icici method...
Around concern Before calling actual method
icici bank interest rate
Around Concern After calling actual method
Calling pnb method...
Around concern Before calling actual method
pnb bank interest rate
Around Concern After calling actual method
```

Problem Statement 4 : Write a program to demonstrate Spring AOP – after returning advice.
Solution : Bankaopdata1.java

```
package Sample;
import org.aspectj.lang.JoinPoint;
import org.aspectj.lang.ProceedingJoinPoint; import org.aspectj.lang.annotation.AfterReturning; import
org.aspectj.lang.annotation.Around;
import org.aspectj.lang.annotation.Aspect; import org.aspectj.lang.annotation.Pointcut; @Aspect
public class Bankaopdata1 { @AfterReturning(
pointcut ="execution(* Bank.*(..))", returning="result")
}
```

Bank1.java

```
package Sample; public class Bank1
{
public void welcome()
{
System.out.println("welcome to bank");
}
public int icici()
{
System.out.println("icici bank interest rate"); return 7;
}
public int pnb()
{
System.out.println("pnb bank interest rate"); return 6;
} }
```

Bankaopdata1.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
<bean id="opBean" class="Sample.Bank"> </bean>
<bean id="trackMyBean" class="Sample.Bankaopdata"></bean>
<bean class="org.springframework.aop.aspectj.annotation.AnnotationAwareAspectJAutoProxyCrea tor
"></bean>
</beans>
```

Banktest1.java

```
package Sample;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext; public class Banktest1
{
private static ApplicationContext context; public static void main(String[] args)
{
System.out.println("Pranali B36");
context = new ClassPathXmlApplicationContext("Bankaopdata1.xml"); Bank e =(Bank)
context.getBean("opBean");
//System.out.println("Calling welcome method..."); e.welcome();
//System.out.println("Calling icici method..."); e.icici();
//System.out.println("Calling pnb method..."); e.pnb();
} }
```

OUTPUT :

```
<terminated> Banktest (5) [AspectJ/Java Application] C:\java\New folder\bin\javaw.exe (28-Nov-2(
Pranali B36
welcome to bank
AfterReturning concern
Result in advicenull
icici bank interest rate
AfterReturning concern
Result in advice7
pnb bank interest rate
AfterReturning concern
Result in advice6
```

Problem Statement 5 : Write a program to demonstrate Spring AOP – after throwing advice.
Solution :

Operationaop_at.java

```
package Sample;
import org.aspectj.lang.JoinPoint;
import org.aspectj.lang.annotation.AfterThrowing; import org.aspectj.lang.annotation.Aspect; @Aspect
public class Operationaop_at { @AfterThrowing(
pointcut = "execution(* Operation_at.*(..))", throwing = "error") public void myadvice(JoinPoint jp,
Throwable error)
{
System.out.println("AfterThrowing concern"); System.out.println("Exception is: "+error);
System.out.println("end of after throwing advice      ");
}
}
```

Operation_at.java

```
package Sample;
public class Operation_at
{
public void validate(int att)throws Exception
{
if(att<75)
{
}                                throw new ArithmeticException("Not eligible for exam");
else
{
}                                System.out.println("Eligible for exam");
}
}
```

validctx.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
<bean id="opBean" class="Sample.Operation_at"></bean>
<bean id="trackMyBean" class="Sample.Operationaop_at"></bean>
<bean class="org.springframework.aop.aspectj.annotation.AnnotationAwareAspectJAutoProxyCrea tor
">

</bean>
</beans>
```

OperationTest_at.java

```
package Sample;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext; public class
OperationTest_at
{
private static ApplicationContext context; public static void main(String[] args)
{

System.out.println("Pranali B36");
ApplicationContext context = new ClassPathXmlApplicationContext("validctx.xml");
Operation_at op = (Operation_at) context.getBean("opBean"); System.out.println("calling validate
        ");
try
{
op.validate(85);
}catch(Exception e)
{
System.out.println(e);
}
System.out.println("calling validate again.    ");
try
{
op.validate(25);
}catch(Exception e)
{
System.out.println(e);
}
}
}
```

Output:

```
<terminated> TestValidation (0) [AspectJ/Java Application] C:\java\New Folder\bin\javaw.exe (20-Nov
Pranali B36
calling validate....
Eligible for exam
calling validate again....
AfterThrowing concern
Exception is: java.lang.ArithmeticException: Not eligible for exam
end of after throwing advice....
java.lang.ArithmeticException: Not eligible for exam
```

Problem Statements 6: Write a program to demonstrate Spring AOP –pointcuts.

Operation_pc.java

```
package Sample;
public class Operation_pc { public void msg()
{
System.out.println("method 1");
}
public int m()
{
System.out.println("method 2 with return"); return 2;
}
public int k()
{
System.out.println("method 3 with return"); return 3;
}
}
```

Aopdata_pc.java

```
package Sample;
import org.aspectj.lang.JoinPoint; import org.aspectj.lang.annotation.After;
import org.aspectj.lang.annotation.Pointcut; import org.aspectj.lang.annotation.Aspect; import
org.aspectj.lang.annotation.Before; @Aspect
public class Aopdata_pc
{
@Pointcut("execution(int Operation.*(..))") public void p(){}
@After("p()")
public void myadvice(JoinPoint jp)
{
System.out.println("After advice");
}
@Pointcut("execution(* Operation.*(..))") public void i(){}
@Before("i()")
public void myadvice1(JoinPoint jp)
{
System.out.println("Before advice");
}
}
```

Test_pc.java

```
package Sample;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext; public class Test_pc {
public static void main(String[] args) { System.out.println("Pranali B36");
ApplicationContext context = new ClassPathXmlApplicationContext("aopctx_pc.xml"); Operation_pc
e=(Operation_pc)context.getBean("opBean");
System.out.println("calling m1..."); e.msg(); System.out.println("calling m2..."); e.m();
System.out.println("calling m3..."); e.k();
}
}
```

aopctx_pc.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
<bean id="opBean" class="Sample.Operation_pc"> </bean>
<bean id="trackMyBean" class="Sample.Aopdata_pc"></bean>

<bean class="org.springframework.aop.aspectj.annotation.AnnotationAwareAspectJAutoProxyCrea tor
"></bean>
</beans>
```

Output:

```
<terminated> Test_pc (2) [AspectJ/Java Application] C:
Pranali B36
calling m1...
method 1
calling m2...
method 2 with return
calling m3...
method 3 with return
```

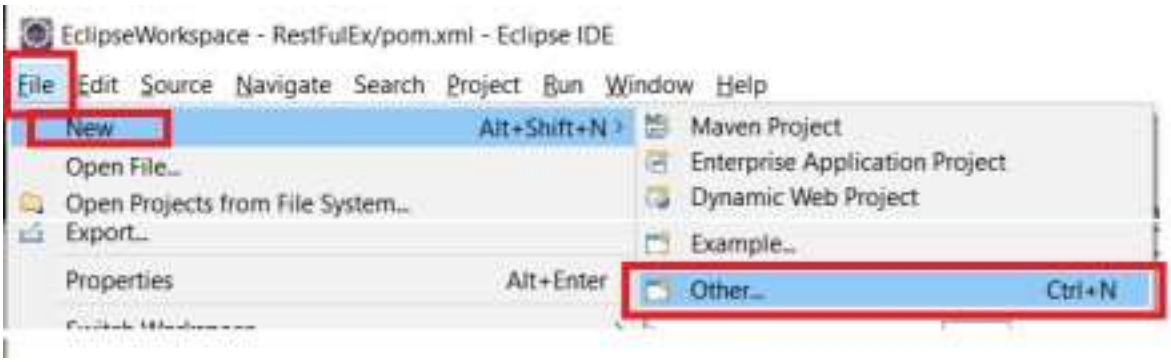
PRACTICAL NO : 9

Spring JDBC

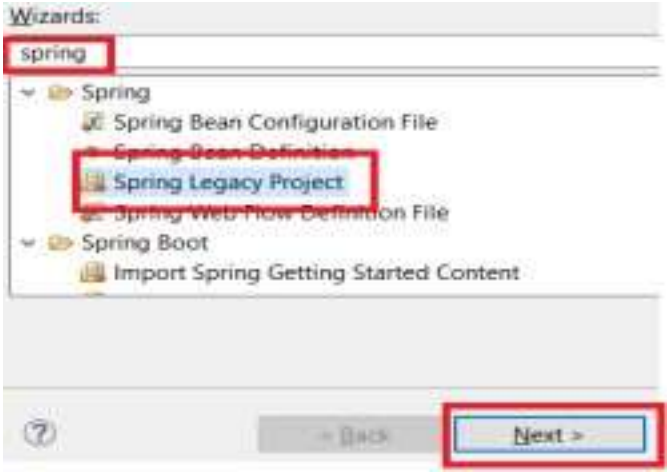
Steps to Create Spring Legacy Project

Step 1 : Creating Spring Legacy Project.

1.1 : Open Eclipse. Go To File > New > Other.

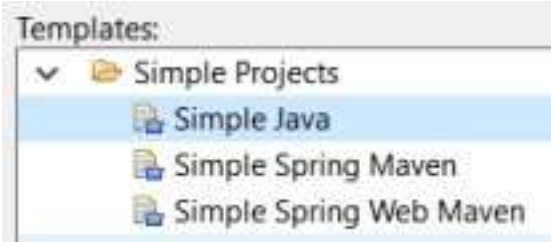


1.2 : Search for 'spring' and Select 'Spring Legacy Project'. Then Click on

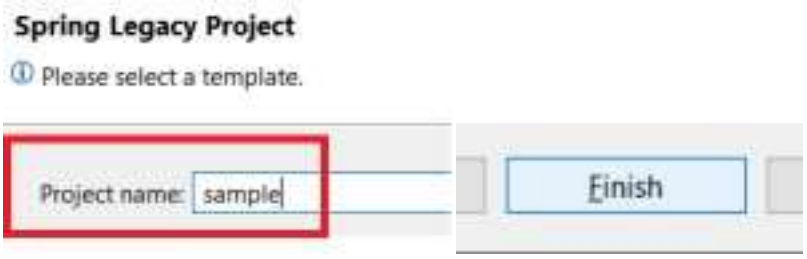


Next.

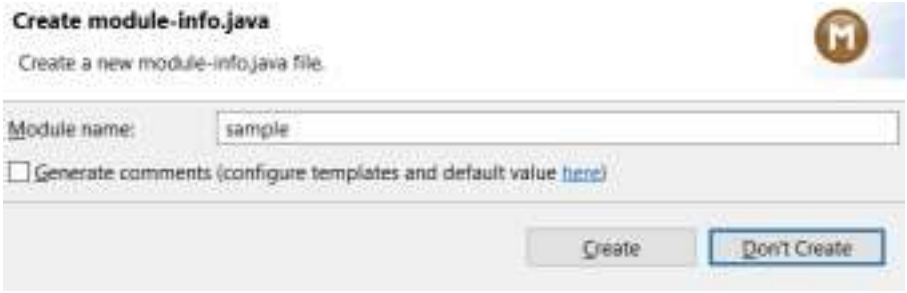
1.3 : ChooseProject Name of your wish, below there select **Simple Java**&



simply Finish.



1.4 : If asked for Creating module-info.java file, click on **Don't Create**.

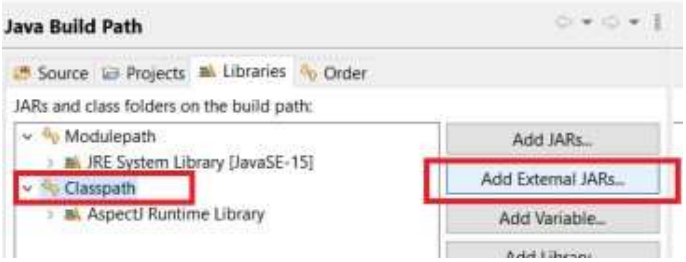


Step 2 : Adding the Spring Libraries.

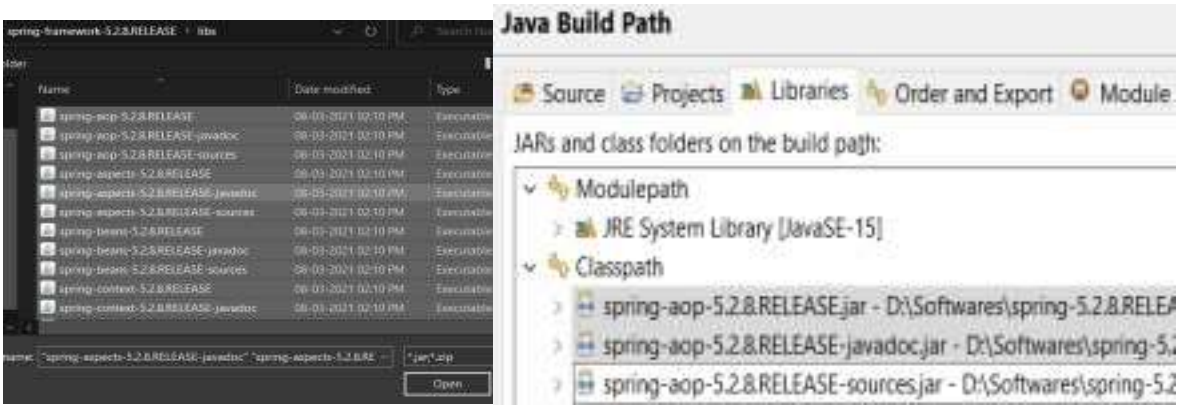
2.1 : Right click on your Newly created Spring Legacy project, Choose Build Path > Configure Build Path.

2.2

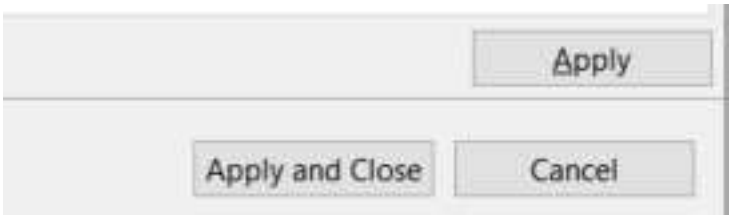
On Java Build Path wizard, Choose **Classpath** and then select **Add ExternalJARs.**



2.3 : Choose all the Spring Libraries you've downloaded, and click on OPEN. This will add all libraries to Classpath.



2.4 Finally click on Apply & Close, now you are ready to work with Spring



Legacy Project.

1. Write a program to insert, update and delete records from the given table. Code:

Movie1.java **package**

```
com.abc; public class
Movie1 { int mid;
String title; String actor;
public Movie1(int mid, String title, String actor) { super();
this.mid = mid; this.title =
title; this.actor = actor;
}
public Movie1() {
super(); }
public int getMid() { return
mid;
}
public void setMid(int mid) { this.mid =
mid;
}
public String getTitle() { return title;
}
public void setTitle(String title) { this.title =
title;
}
public String getActor() { return
actor;
}
public void setActor(String actor) { this.actor =
actor;
}
}
```

MovieDAO.java

```
package com.abc; import
org.springframework.jdbc.core.*; public class
MovieDAO { JdbcTemplate jdbcTemplate;
public void setJdbcTemplate(JdbcTemplate jdbcTemplate) { this.jdbcTemplate =
jdbcTemplate;
} public int insMovie(Movie1 m1)
{
String insSql="insert into mymovies1
values("+m1.getMid()+",'"+m1.getTitle()+"','"+m1.getActor()+"')";

return jdbcTemplate.update(insSql);
} public int updateMovie(Movie1 m1){
String query="update mymovies1 set
title='"+m1.getTitle()+"',actor='"+m1.getActor()+"'where mid='"+m1.getMid()+"' "; return
jdbcTemplate.update(query);
} public int deleteMovie(Movie1 m1){
String query="delete from mymovies1 where mid='"+m1.getMid()+"' "; return
jdbcTemplate.update(query);
}
}
```

MovieTest.java package

```
com.abc;
```

```
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext; public class
MovieTest {
private static ApplicationContext appCon; public static void
main(String[] args) { // TODO Auto-generated method stub appCon =
new ClassPathXmlApplicationContext("appctx.xml");
MovieDAO m1 = (MovieDAO) appCon.getBean("mymovie");
// insert query
Movie1 t1 = new Movie1(1, "Shiddat", "Saloni");
System.out.println(m1.insMovie(t1));
Movie1 t2 = new Movie1(2, "YJHD", "Siddhika");
System.out.println(m1.insMovie(t2));
Movie1 t3 = new Movie1(3, "Sairat", "Priyanka");
System.out.println(m1.insMovie(t3));
 //update query int status = m1.updateMovie(new Movie1(1, "3idiots",
"Saloni")); System.out.println(status);
// delete
Movie1 t2=new Movie1();
t2.setMid(2); int
s=m1.deleteMovie(t2);
System.out.println(s);}}
```

Appctx.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
<bean id="ds" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
<property name="driverClassName" value="org.postgresql.Driver" />
<property name="url" value="jdbc:postgresql://localhost:5432/postgres" />
<property name="username" value="postgres" />
<property name="password" value="root" />
</bean>
<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
<property name="dataSource" ref="ds"></property>
</bean>
<bean id="mymovie" class="com.abc.MovieDAO">
<property name="jdbcTemplate" ref="jdbcTemplate"></property>
</bean> </beans>
```

Output:

```
CREATE TABLE mymovies1 (
mid int,
title varchar(50),
actor varchar(50),
PRIMARY KEY (mid)
);
```

Insert

Markers Properties Servers Data Source Explo

<terminated> MovieTest [Java Application] C:\Program Files

1
1
1

	mid [PK] integer	title character varying (50)	actor character varying (50)
1	1	Shiddat	Saloni
2	2	YJHD	Siddhika
3	3	Sairat	Priyanka

Update

Markers Properties Servers Data Sourc

<terminated> MovieTest [Java Application] C:\Progra

1

	mid [PK] integer	title character varying (50)	actor character varying (50)
1	2	YJHD	Siddhika
2	3	Sairat	Priyanka
3	1	3idiots	Saloni

Delete

Markers Properties Servers Data Sour

<terminated> MovieTest [Java Application] C:\Progr

1

	mid [PK] integer	title character varying (50)	actor character varying (50)
1	3	Sairat	Priyanka
2	1	3idiots	Saloni

2. **Write a program to demonstrate PreparedStatement in Spring JdbcTemplate.** Code:

Movie1.java **package**

```
com.abc; public class
Movie1 { int mid;
String title; String actor;
public Movie1(int mid, String title, String actor) { super();

this.mid = mid; this.title =
title; this.actor = actor;
}
public Movie1() { super();
// TODO Auto-generated constructor stub
}
public int getMid() { return
mid;
}
public void setMid(int mid) { this.mid =
mid;
}
public String getTitle() { return title;
}
public void setTitle(String title) { this.title =
title;
}
public String getActor() { return
actor;
}
public void setActor(String actor) { this.actor =
actor;
}

}
```

MovieDAO1.java

```
package com.abc; import
java.sql.PreparedStatement;
import java.sql.SQLException;

import org.springframework.dao.DataAccessException; import
org.springframework.jdbc.core.JdbcTemplate; import
org.springframework.jdbc.core.PreparedStatementCallback; public class
MovieDAO1 { JdbcTemplate jdbcTemplate;

public void setJdbcTemplate(JdbcTemplate jdbcTemplate) { this.jdbcTemplate =
jdbcTemplate;
} public Boolean saveMovieByPreparedStatement(final Movie1 e){ String query="insert
into mymovies1 values(?,?,?)"; return jdbcTemplate.execute(query,new
PreparedStatementCallback<Boolean>(){
@Override
public Boolean doInPreparedStatement(PreparedStatement ps) throws
SQLException, DataAccessException {
ps.setInt(1,e.getMid());
ps.setString(2,e.getTitle());
ps.setString(3,e.getActor()); return
ps.execute();
} }); }
```

```
}
```

MovieTest1.java package

```
com.abc;
import org.springframework.context.ApplicationContext; import
org.springframework.context.support.ClassPathXmlApplicationContext; public class
MovieTest1 {

private static ApplicationContext appCon; public static void
main(String[] args) { // TODO Auto-generated method stub appCon =
new ClassPathXmlApplicationContext("appctx.xml"); MovieDAO1
m1=(MovieDAO1)appCon.getBean("mymovie");
m1.saveMovieByPreparedStatement(new Movie1(5,"Bhaijaan","Slemon"));
}
}
```

Appctx.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="ds" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
<property name="driverClassName" value="org.postgresql.Driver" />
<property name="url" value="jdbc:postgresql://localhost:5432/postgres" />
<property name="username" value="postgres" />
<property name="password" value="root" />
</bean>
<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
<property name="dataSource" ref="ds"></property>
</bean>
<bean id="mymovie" class="com.abc.MovieDAO1">
<property name="jdbcTemplate" ref="jdbcTemplate"></property>
</bean> </beans>
```

Output:

Data Output Messages Notifications

	mid [PK] integer	title character varying (50)	actor character varying (50)
1	3	Sairat	Priyanka
2	1	3idiots	Saloni
3	5	Bhaijaan	Slemon

3. Write a program in Spring JDBC to demonstrate ResultSetExtractor Interface.

Code:

Movie2.java **package**

```
com.abc; public class
Movie2 { int mid;
String title; String actor;
public int getMid() {
return mid;
}
public void setMid(int mid) { this.mid =
mid;
}
public String getTitle() { return title;
}
public void setTitle(String title) { this.title =
title;
}
public String getActor() { return
actor;
}
public void setActor(String actor) { this.actor =
actor;
}
public String toString(){ return
mid+" "+title+" "+actor;
}
}
```

MovieDAO2.java package

```
com.abc; import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList; import
java.util.List;
import org.springframework.dao.DataAccessException; import
org.springframework.jdbc.core.JdbcTemplate; import
org.springframework.jdbc.core.ResultSetExtractor;
public class MovieDAO2 { JdbcTemplate
jdbcTemplate;

public void setJdbcTemplate(JdbcTemplate jdbcTemplate) { this.jdbcTemplate =
jdbcTemplate;
}
public List<Movie2> getAllMovie(){
return jdbcTemplate.query("select * from mymovies1",new
ResultSetExtractor<List<Movie2>>(){

@Override
public List<Movie2> extractData(ResultSet rs) throws SQLException,
DataAccessException {

List<Movie2> list=new ArrayList<Movie2>();
while(rs.next()){
Movie2 e=new Movie2(); e.setMid(rs.getInt(1));
e.setTitle(rs.getString(2));
e.setActor(rs.getString(3)); list.add(e);
}
```

```
return list;
} }); }
}
```

MovieTest2.java **package**

```
com.abc; import java.util.List;
import org.springframework.context.ApplicationContext; import
org.springframework.context.support.ClassPathXmlApplicationContext; public class
MovieTest2 {
private static ApplicationContext appCon; public static void
main(String[] args) { appCon = new
ClassPathXmlApplicationContext("appctx.xml");
MovieDAO2 m1=(MovieDAO2)appCon.getBean("mymovie");
List<Movie2> list=m1.getAllMovie();

for(Movie2 e:list)
System.out.println(e);
}
}
```

Appctx.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="ds" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
<property name="driverClassName" value="org.postgresql.Driver" />
<property name="url" value="jdbc:postgresql://localhost:5432/postgres" />
<property name="username" value="postgres" />
<property name="password" value="root" />
</bean>
<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
<property name="dataSource" ref="ds"></property>
</bean>
<bean id="mymovie" class="com.abc.MovieDAO2">
<property name="jdbcTemplate" ref="jdbcTemplate"></property>
</bean>
</beans>
```

Output:

	mid [PK] integer	title character varying (50)	actor character varying (50)
1	3	Sairat	Priyanka
2	1	3idiots	Saloni
3	5	Bhaijaan	Slemon
4	2	YJHD	Siddhika

Markers Properties Servers Data Source Explor

```
<terminated> MovieTest2 [Java Application] C:\Program File
3 Sairat Priyanka
1 3idiots Saloni
5 Bhaijaan Slemon
2 YJHD Siddhika
```

4. Write a program to demonstrate RowMapper interface to fetch the records from the database.

Code:

Movie3.java **package**

```
com.abc; public class
Movie3 { int mid;
String title; String actor;
public Movie3(int mid, String title, String actor) { super();
this.mid = mid; this.title =
title; this.actor = actor;
}

public Movie3() { super();
// TODO Auto-generated constructor stub
}
public int getMid() { return
mid;
}
public void setMid(int mid) { this.mid =
mid;
} public String getTitle() {

return title;
}
public void setTitle(String title) { this.title =
title;
}
public String getActor() { return
actor;
}
public void setActor(String actor) { this.actor =
actor;
}
}
```

MovieDAO3.java

```
package com.abc; import
java.sql.ResultSet; import
java.sql.SQLException;
import java.util.List; import
org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.jdbc.core.RowMapper;

public class MovieDAO3 { JdbcTemplate
jdbcTemplate;

public void setJdbcTemplate(JdbcTemplate jdbcTemplate) { this.jdbcTemplate =
jdbcTemplate;

}
public List<Movie2> getAllEmployeesRowMapper(){ return jdbcTemplate.query("select *
from mymovies1",new RowMapper<Movie2>(){
@Override public Movie2 mapRow(ResultSet rs, int rownumber) throws SQLException
```

```
{ Movie2 e=new Movie2(); e.setMid(rs.getInt(1));
e.setTitle(rs.getString(2));
e.setActor(rs.getString(3));
return e; }
}); }

}
```

MovieTest3.java **package**

```
com.abc; import java.util.List;
import org.springframework.context.ApplicationContext; import
org.springframework.context.support.ClassPathXmlApplicationContext; public class
MovieTest3 {
private static ApplicationContext appCon; public static void
main(String[] args) { // TODO Auto-generated method stub appCon =
new ClassPathXmlApplicationContext("appctx.xml");
MovieDAO3 m1=(MovieDAO3)appCon.getBean("mymovie");
List<Movie2> list=m1.getAllEmployeesRowMapper();

for(Movie2 e:list)
System.out.println(e);
}}
```

Appctx.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="ds" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
<property name="driverClassName" value="org.postgresql.Driver" />
<property name="url" value="jdbc:postgresql://localhost:5432/postgres" />
<property name="username" value="postgres" />
<property name="password" value="root" />
</bean>
<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
<property name="dataSource" ref="ds"></property>
</bean>
<bean id="mymovie" class="com.abc.MovieDAO3">
<property name="jdbcTemplate" ref="jdbcTemplate"></property>
</bean>
</beans>
```

Output:

PRACTICAL NO : 10