Assignments on Java Generics

1. Write a Java Program to demonstrate a Generic Class.

```
 class G1<T>
{
	T obj;
	G1(T obj)
	{
		this.obj = obj;
	}
	public T get()
	{
		return (this.obj);
	}
}

public class Generic
{
	public static void main(String[] args) {
	// TODO Auto-generated method stub

		G1 <String> s = new G1 <String>("Atharv");
		System.out.println(s.get());

		G1 <Integer> i =new G1 <Integer>(1);
		System.out.println(i.get());

		G1 <Double> d = new G1<Double>(11.11);

				System.out.println(d.get());
			}
	}
```

OUTPUT :-

```
Markers  Properties
<terminated> Generic [Java A
Atharv
1
11.11
```

A) Parameterized type

```
class test_gp <T1,T2>
{
	T1 obj1;
	T2 obj2;
	test_gp(T1 obj1, T2 obj2)
{
	this.obj1 = obj1;
	this.obj2 = obj2;
}

	public void print()
	{
		System.out.println("T1 Object : " + obj1);
		System.out.println("T2 Object : " + obj2);

	}
}

public class gpClass {


	public static void main(String[] args) {
		// TODO Auto-generated method stub

		test_gp<String, Integer>obj= new test_gp <String, Integer> ("Parameterized
Generics", 11);

		obj.print();

	}

}
```

OUTPUT :-

```
Markers  Properties  Servers  Data Source Explorer  Snippe
<terminated> gpClass [Java Application] C:\Program Files\Java\jdk-11.0.16\bin\
T1 Object : Parameterized Generics
T2 Object : 11
```

2. Write a Java Program to demonstrate Generic Methods.

```
class gMethod {
	static <T> void gDisplay (T e)
	{
		System.out.println(e.getClass().getName() + " = " + e);
	}

	public static void main(String[] args) {
		// TODO Auto-generated method stub

		gDisplay(1);
		gDisplay("Atharv");
		gDisplay("11.11");

	}

}
```

OUTPUT :-

```
Markers  Properties  Servers  Data Source Explorer
<terminated> gMethod [Java Application] C:\Program Files\Java\jdk-11
java.lang.Integer = 1
java.lang.String = Atharv
java.lang.String = 11.11
```

3. Write a Java Program to demonstrate Wildcards in Java Generics.

UPPER BOUNDED WILDCARDS

```
import java.util.Arrays;
import java.util.List;

class ubWildcard {

        private static double sum(List <? extends Number> list)
        {
        double sum=0.0;
        for (Number i: list)
        {
        sum+=i.doubleValue();
        }
        return sum; }


    public static void main(String[] args)
    {

    List<Integer> l1= Arrays.asList(4,5,6,7);
    System.out.println("Total sum is:"+sum(l1));

    List<Double> l2=Arrays.asList(4.1,5.1,6.1);
    System.out.println("Total sum is : "+sum(l2));
    }
}
```

OUTPUT :-

```
Markers  Properties  Servers  Data Source Explorer  Sni
<terminated> ubWildcard [Java Application] C:\Program Files\Java\jdk-11.(
Total sum is:22.0
Total sum is : 15.299999999999999
```

LOWER BOUNDED WILDCARDS

```
        import java.util.Arrays;

        import java.util.List;

class lbWildcard {

        public static void print1(List<?super Integer> list)

        {
        System.out.println(list);
        }

        public static void main(String[] args)
        {
        List<Integer> list1= Arrays.asList(4,5,6,7);

        print1(list1);
        List<Number> list2 = Arrays.asList(4,5,6,7);
        print1(list2);

        }
}
```

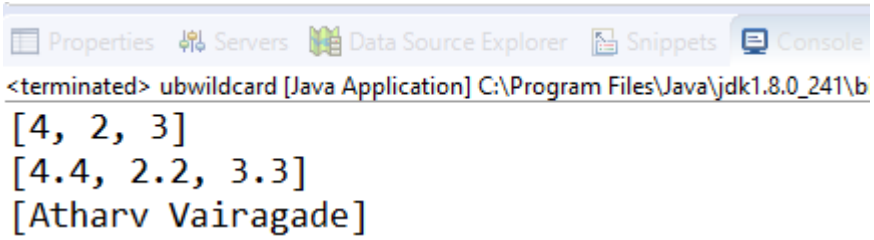
OUTPUT :-

```
Markers  Properties  Serve
<terminated> lbWildcard [Java Applica
[4, 5, 6, 7]
[4, 5, 6, 7]
```

UNBOUNDED WILDCARDS

```
package java_wc;
import java.util.Arrays;
import java.util.List;

class ubwildcard {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		List<Integer> list1 = Arrays.asList(4, 2, 3);
		List<Double> list2 = Arrays.asList(4.4, 2.2, 3.3);
		List<String> list3 = Arrays.asList("Atharv Vairagade");

		printlist(list1);
		printlist(list2);
		printlist(list3);

	}
	private static void printlist(List<?> list)
	{
		System.out.println(list);
	}
}
```

OUTPUT :-



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.**

Solution :-

```
package java_wc;
        import java.util.*;
        public class list {
                public static void main(String[] args) {
                int a;
                ArrayList<String> list=new ArrayList<String>();
                list.add("One");
                list.add("Two");
                list.add("Three");
                list.add("Four");
                System.out.println(list);
                System.out.println("Traversing list through for each loop ");
                for(String number:list)
                System.out.println(number);

                System.out.println("Traversing list through Iterator ");
                Iterator itr=list.iterator();
                while(itr.hasNext()){

                System.out.println(itr.next());

                        }
                }
        }
```

OUTPUT :-

```
Properties  Servers  Data Source Explorer  Snippets  C
<terminated> list [Java Application] C:\Program Files\Java\jdk1.8.0_241\bi
[One, Two, Three, Four]
Traversing list through for each loop
One
Two
Three
Four
Traversing list through Iterator
One
Two
Three
Four
```

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.

Solution :-

```
package atharv;
import java.util.*;

class ListIteratorExample {

   public static void main(String[] args) {

      List<String> items = new ArrayList<>();
      items.add("Atharv");
      items.add("Safal");
      items.add("Yash");
      items.add("Sujal");
      items.add("Suraj");

      ListIterator<String> listIterator = items.listIterator();

            System.out.println("Items in forward direction:");
      while (listIterator.hasNext()) {
         System.out.println(listIterator.next());
      }

            System.out.println("\nItems in backward direction:");
      while (listIterator.hasPrevious()) {
         System.out.println(listIterator.previous());
      }
   }
}
```

OUTPUT :-

```
<terminated> listitr [Java Application] C:\
Items in forward direction:
Atharv
Safal
Yash
Sujal
Suraj

Items in backward direction:
Suraj
Sujal
Yash
Safal
Atharv
```

Assignments on 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 / backward direction.

Solution :-

```
package atharv;

        import java.util.*;
        class hashset {


                public static void main(String[] args)
                {
                Set<String> h = new HashSet<String>();

                h.add("Atharv");
                h.add("Yash");
                h.add("Safal");

                System.out.println(h);

                System.out.println("Iterating over set:");

                Iterator<String> i = h.iterator();
                while (i.hasNext())
                System.out.println(i.next());

                 List<String> itemList = new ArrayList<>(h);

            Collections.reverse(itemList);

            System.out.println("\nItems in reverse order:");
            for (String item : itemList) {
               System.out.println(item);
                    }
              }
        }
```

OUTPUT :-

```
Markers  Properties  Servers
<terminated> hashset [Java Application] C:
Iterating over set:
Yash
Safal
Atharv

Items in reverse order:
Atharv
Safal
Yash
```

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 into other set.
c. Remove items from the set
d. Search the specified item in the set

Solution :-

```
Package Atharv;
import java.util.Arrays;
import java.util.HashSet;
import java.util.Set;

public class SetOperationsWithArray {
 public static void main(String[] args) {
 // Create an array of items
        String[] itemsArray = {"Apple", "Banana", "Orange", "Mango"};

 // a. Add items in the set using an array
 Set<String> set1 = new HashSet<>(Arrays.asList(itemsArray));

        System.out.println("Set 1 after adding items from array: " + set1);

 // b. Insert items of one set into another set
 String[] newItemsArray = {"Grapes", "Pineapple"};

        Set<String> set2 = new HashSet<>(Arrays.asList(newItemsArray));
        System.out.println("Set 2 before insertion: " + set2);

        // Insert set1 into set2
        set2.addAll(set1);
        System.out.println("Set 2 after inserting Set 1: " + set2);

        // c. Remove items from the set
        set2.remove("Banana");

        System.out.println("Set 2 after removing 'Banana': " + set2);

        // d. Search the specified item in the set using an array
        String[] searchItems = {"Orange", "Strawberry"};
        for (String item : searchItems) {
```

```
 if (set2.contains(item)) {

 System.out.println(item + " is found in the set.");
 }
else
{
 System.out.println(item + " is not found in the set.");
            }
        }
    }
}
```

OUTPUT :-

```
Output
java -cp /tmp/MpAe81oG7g/SetOperationsWithArray
Set 1 after adding items from array: [Apple, Mango, Orange, Banana]
Set 2 before insertion: [Grapes, Pineapple]
Set 2 after inserting Set 1: [Apple, Grapes, Mango, Pineapple, Orange, Banana]
Set 2 after removing 'Banana': [Apple, Grapes, Mango, Pineapple, Orange]
Orange is found in the set.
Strawberry is not found in the set.

=== Code Execution Successful ===
```

Assignments on 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.

Solution :-

```
Package atharv;
        import java.util.*;
        public class mapdemo {

public static void main(String[] args) {
Map<String, String> hmap=new HashMap<>();
                hmap.put("India", "New Delhi");
                hmap.put("South Korea", "Seoul");
                hmap.put("Japan", "Tokyo");
                hmap.put("Russia", "Moscow");
                hmap.put("UK", "London");
for(Map.Entry m:hmap.entrySet())

{
System.out.println("capital of "+m.getKey()+" is "+m.getValue());
}
System.out.println("----------");
hmap.remove("UK");
for(Map.Entry m:hmap.entrySet()){
System.out.println("capital of "+m.getKey()+" is "+m.getValue());

}
System.out.println("----------");
System.out.println("capital of India is "+hmap.get("India"));
System.out.println("----------");
        Map<String, String> hmap2=new HashMap<>();
        hmap2.put("Germany", "Berlin");
```

```
        hmap2.put("Georgia", "Tbilisi");
        hmap.putAll(hmap2);
        for(Map.Entry m:hmap.entrySet()) {
System.out.println("capital of "+m.getKey()+" is "+m.getValue());
            }
        }
}
```

OUTPUT:-

```
Output
java -cp /tmp/NIECfOlzIz/mapdemo
capital of South Korea is Seoul
capital of Japan is Tokyo
capital of UK is London
capital of India is New Delhi
capital of Russia is Moscow
----------
capital of South Korea is Seoul
capital of Japan is Tokyo
capital of India is New Delhi
capital of Russia is Moscow
----------
capital of India is New Delhi
----------
capital of South Korea is Seoul
capital of Japan is Tokyo
capital of Georgia is Tbilisi
capital of Germany is Berlin
capital of India is New Delhi
capital of Russia is Moscow
```

Assignments on Lambda Expression

1. **Write a Java program using Lambda Expression to print “Hello World”.**

Solution:

```
package lambdaav;
interface Helloworld{
	public void sayhello();
}
public class helloworld {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
Helloworld h1 = () -> {
	System.out.println("Hello, atharv");
};
h1.sayhello();
	}

}
```

OUTPUT :-

```
Problems   @ Javadoc   Console   helloworld.java   SqArea.java
<terminated> helloworld [Java Application] C:\Program Files\Java\jdk-11.0.16\bin\javaw.exe (10-Sep-2024, 11:57:55 am)
Hello, atharv
```

2. **Write a Java program using Lambda Expression with a single parameter.**

Solution:

```
interface SquareArea {
public double sqArea(double s);
}
        public class Area
        {
                public static void main (String[] args)
                {
        SquareArea sq = (double s) -> { return s*s
        };
         System.out.println("Atharv Vairagade");
        System.out.println("Area of Square with side 9.9 is : " + sq.sqArea(9.9));
                }
        }
```

OUTPUT :-

```
Console
<terminated> Area (1) [Java Application] C:\Program F
Atharv Vairagade
Area of Square with side 9.9 is : 98.01
```

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

Solution:

```
public interface sum{
public int sumint(int a, int b);
}
public class sum {
        public static void main(string[] args)
                {
                        sum s1 = (int a, int b) ->
                                {
                                        return a+b;
                                };
                system.out.println("Atharv Vairagade");
                system.out.println("1 + 3 = " + s1.sumint(1,3));
                system.out.println("5 + 8 = " + s1.sumint(5,8));
        }
}
```

OUTPUT :-

```
Console
<terminated> sum [Java Application] C:\Program
Atharv Vairagade
1 + 3 = 4
5 + 8 = 13
```

4. Write a Java program using Lambda Expression to calculate the following:
a. Convert Fahrenheit to Celsius.
b. Convert Kilometres to Miles.

Solution:

```
interface TempConversion {
public double convert(double f);
}
interface DistanceConversion {
public double convert(double km);
}
public class Temperature {

	public static void main(String[] args)
	{
		TempConversion tc = (double f)->(f-32)*5/9;
		System.out.println("-8.4 Farenheit is " +
		String.format("%.2f", tc.convert(-8.4)) + " Celcius");

		System.out.println("32 Farenheit is " + String.format("%.2f",

		tc.convert(32)) + " Celcius");

		DistanceConversion dc = (double km)-> km/1.609344;
		System.out.println("1.5 Kilometre is " +
		String.format("%.2f", dc.convert(1.5)) + " Miles");
	System.out.println("3.28 Kilometre is " +
			String.format("%.2f", dc.convert(3.28)) + " Miles");
	}
}
```

OUTPUT :-

```
-8.4 Farenheit is -22.44 Celcius
32 Farenheit is 0.00 Celcius
1.5 Kilometre is 0.93 Miles
3.28 Kilometre is 2.04 Miles
```

5. Write a Java program using Lambda Expression with or without return Keyword.

Solution:

Without return keyword

```
interface CircleArea {
public double area(double r);
}
public class noReturn {
public static void main(String[] args)
{
CircleArea c1 = (r) -> (3.1415*r*r);

System.out.println("Atharv Vairagade");
System.out.println("Area of circle with radius 90.5 is : " + c1.area(90.5));
}
}
```

With return keyword

```
interface CircleArea {
public double area(double r);
}
public class noReturn {
public static void main(String[] args)
{
CircleArea c1 = (r) -> { return (3.1415*r*r);
};

System.out.println("Atharv Vairagade");
System.out.println("Area of circle with radius 90.5 is : " + c1.area(90.5));
}
}
```

OUTPUT :-

```
Console
<terminated> noReturn [Java Application] C:\Program Files\Java\jre1.8.0_241
Atharv Vairagade
Area of circle with radius 90.5 is : 25729.670374999998
```

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

Solution :-

```
package java_wc;

 interface concateStr {
public String concat(String s1, String s2);
}
public class concatSTR
 {
        public static void main(String[] args)
        {
                concateStr cs = (String s1, String s2) -> s1 + s2;
                System.out.println("concatenating \"Hello,\" and \" world\" = "
                + cs.concat("Hello"," world"));
        }
 }
```

OUTPUT :-

```
Properties  Servers  Data Source Explorer  Snippets  Console
<terminated> concatSTR [Java Application] C:\Program Files\Java\jdk1.8.0_241\bin\javaw.exe (Oct 10, 2024, 4:1
concatenating "Hello," and " world" = Hello, world
```

Web application development using JSP

Problem Statement 6.1 : Create a Telephone directory using JSP and store all the information within a database, so that later could be retrieved as per the requirement.
Solution :

Filename- 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="styl
esheet"
integrity="sha384-BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRu
Wlolflfl"crossorigin="anonymous">
<!-- 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/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1g
YU5S9FOnJ0"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 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: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 TeleDir Where id = " +
request.getParameter("del"));
                        response.sendRedirect("index.jsp");
                }
                 String myDataField =null;
                 String myQuery ="SELECT * FROM TeleDir 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(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></center></body></html>
```

Filename- add.jsp

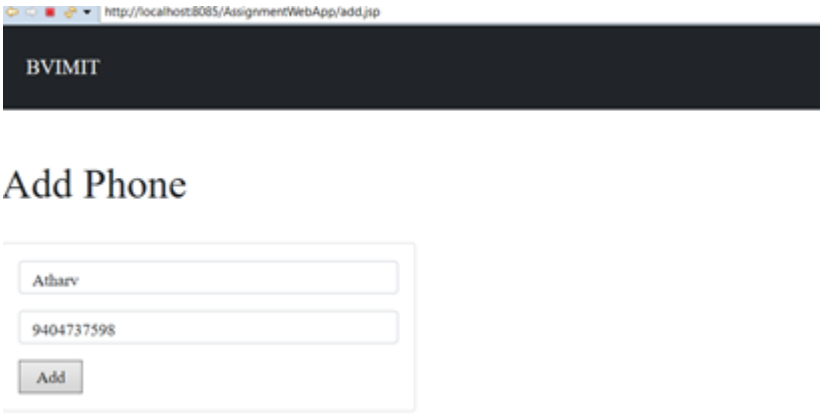
```
<%@page import="java.sql.*"%>
<%@ page language="java"contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
```

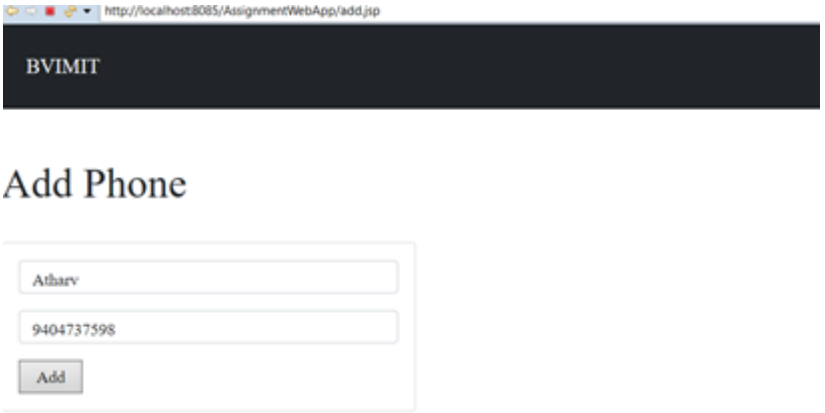
```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Add</title>
<!-- CSS only -->
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.4.0/font/bootstrap-icons.css">
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css"rel="styl
esheet"
integrity="sha384-BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRu
Wlolflfl"crossorigin="anonymous">
<!-- 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/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1g
YU5S9FOnJ0"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 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>
```

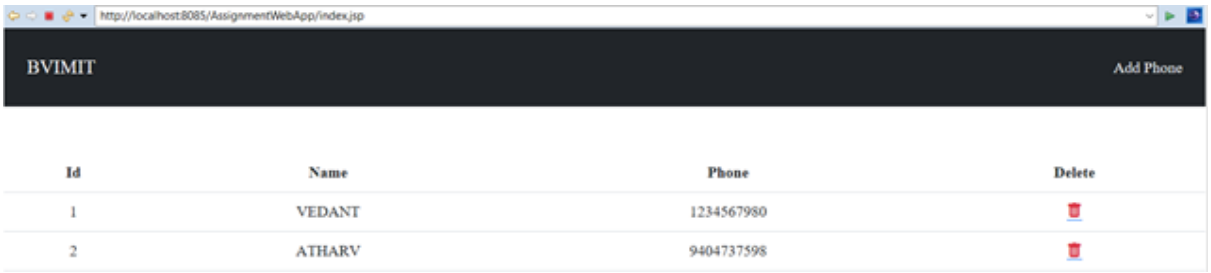
```
<div class="form-group m-2">
                                <input class="btnbtn-primary px-3" type="submit"
value="Add"/>
                        </div>

                </form>


                <%
                try{
                        String driver ="org.postgresql.Driver";
                        String url ="jdbc:postgresql://localhost:5434/postgres";
                        String username ="postgres";
                        String password ="ravita123";
                 Connection con=null;
                Class.forName(driver).newInstance();
                        con = DriverManager.getConnection(url,username,password);
                        if(request.getParameter("phone") != null){
                                PreparedStatementps = 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);
                }               %>
        </center>
</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.
```

Problem Statement 6.2 : Write a JSP page to display the Registration form
Solution :
register_1.jsp

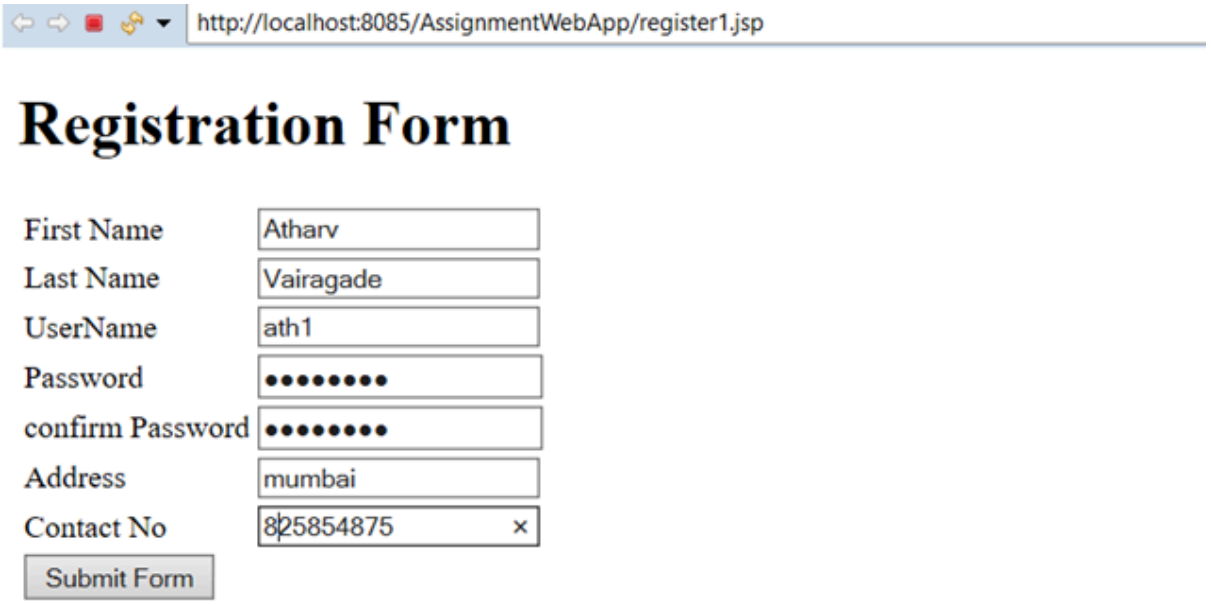
```
<%@ page language="java"contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Registration Form</title>
</head>
<body>
<center>
<h1>Registration Form</h1>

<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>
```

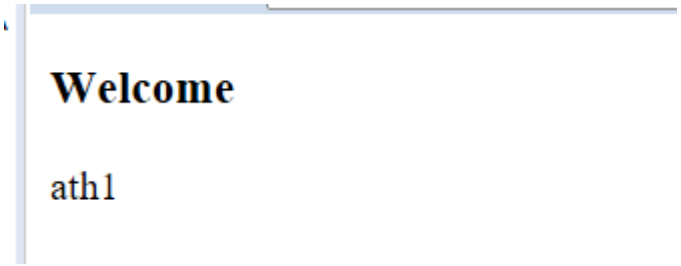
register_2.jsp

```
<%@ 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>
```

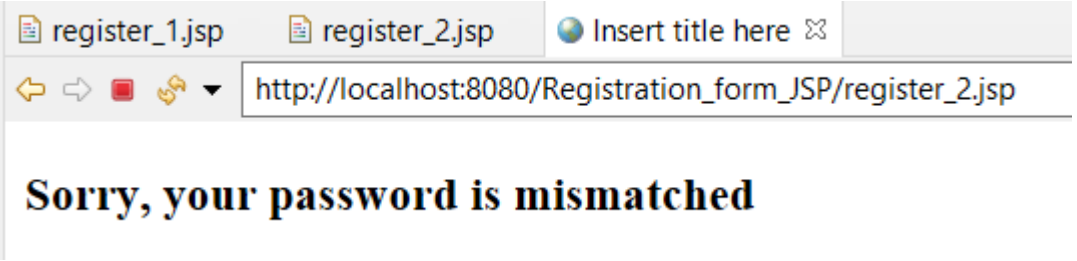
OUTPUT-



After Click on Submit Form



If password is wrong.


register_1.jsp
register_2.jsp
Insert title here
http://localhost:8080/Registration_form_JSP/register_2.jsp
Sorry, your password is mismatched

Problem Statement 6.3 : Write a JSP program to add, delete and display the records from StudentMaster (RollNo, Name, Semester, Course) table.

Solution :

Filename-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="styl
esheet"
integrity="sha384-BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRu
Wlolflfl"crossorigin="anonymous">
<!-- 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/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1g
YU5S9FOnJ0"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="styl
esheet"
integrity="sha384-BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRu
Wlolflfl"crossorigin="anonymous">
<!-- 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/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1g
YU5S9FOnJ0"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

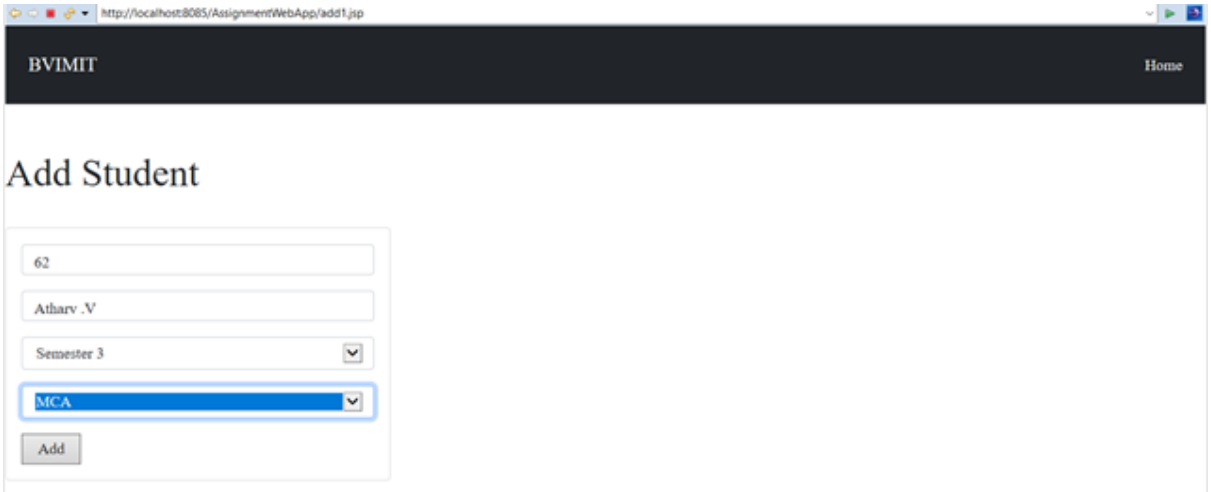
```
<%@page import="java.sql.PreparedStatement"%>
<%@page import="java.sql.Connection"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.Statement"%>
<%@ page language="java"contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!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="styl
esheet"
integrity="sha384-BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRu
Wlolflfl"crossorigin="anonymous">
<!-- 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/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1g
YU5S9FOnJ0"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">
```

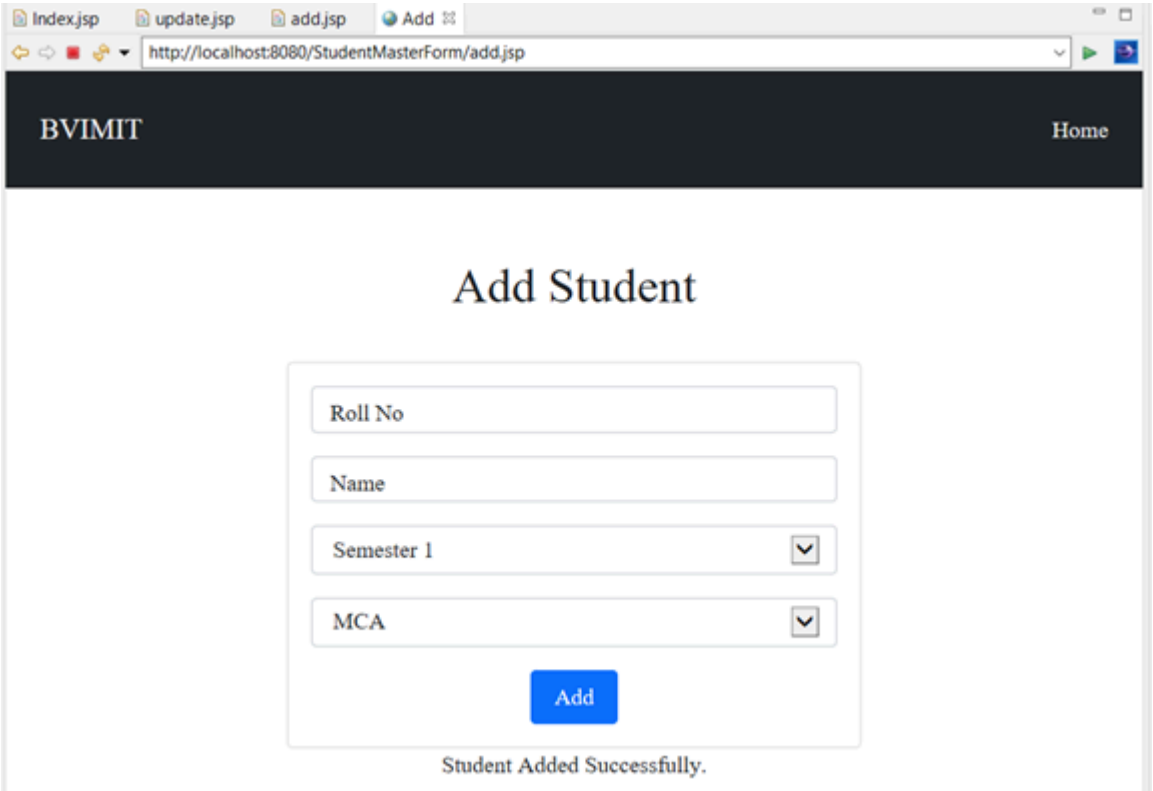
```
                        <option selected disabled="disabled" value="<%=
rs.getString(4)%>"><%= rs.getString(4)%></option>
                        <option value="Sem1">Semester 1</option>
                        <option value="Sem2">Semester 2</option>
                        <option value="Sem3">Semester 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"));
```

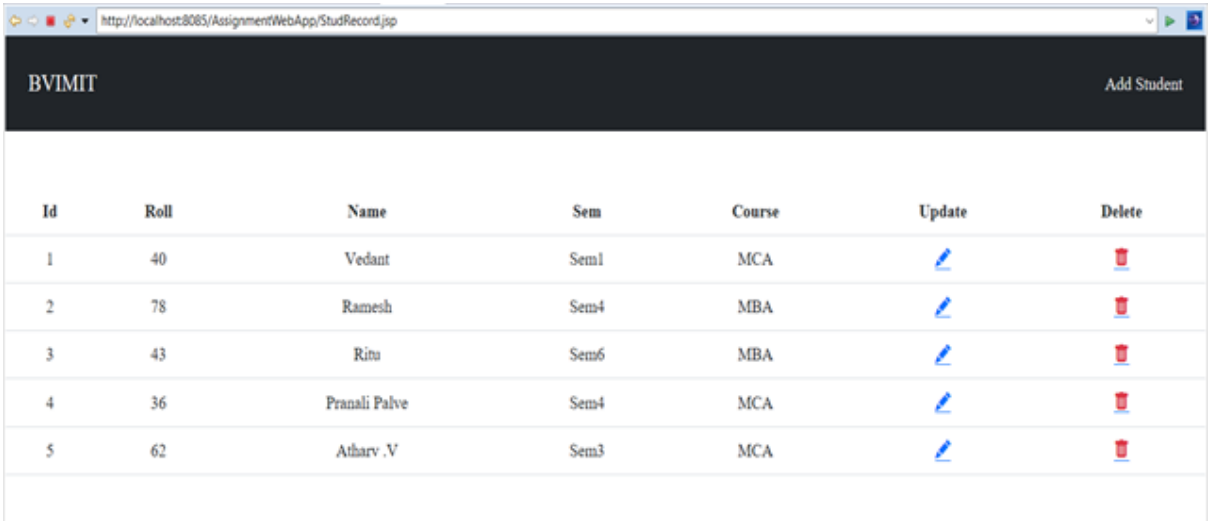
```
String query = "";
PreparedStatementps = null;

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>
```



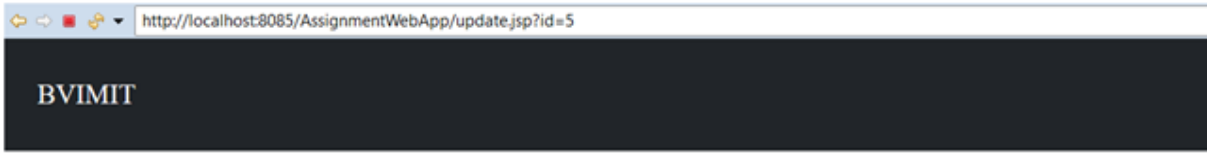
After click on Add

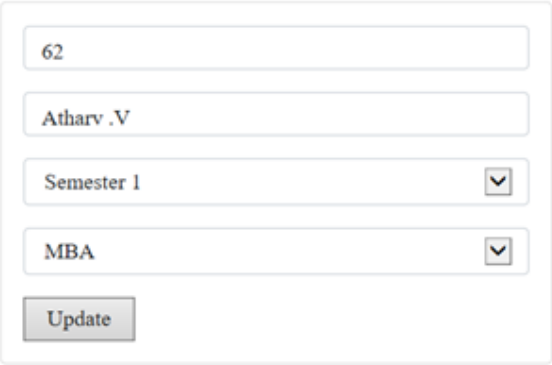


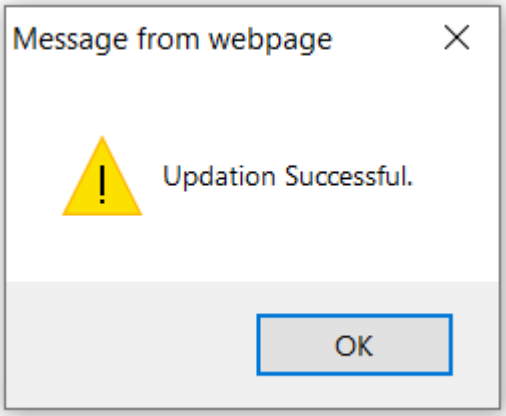


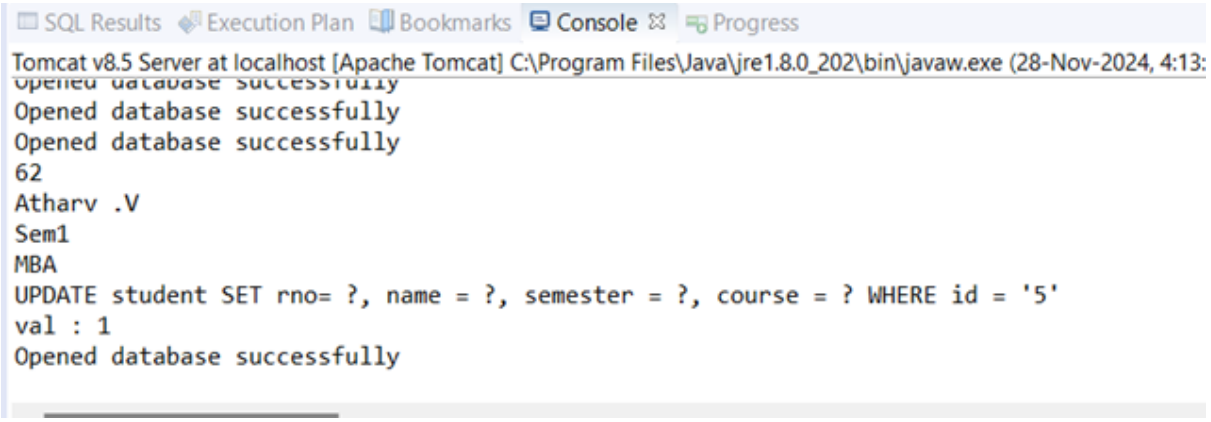
Id	Roll	Name	Sem	Course	Update	Delete
1	40	Vedant	Sem1	MCA		
2	78	Ramesh	Sem4	MBA		
3	43	Ritu	Sem6	MBA		
4	36	Pranali Palve	Sem4	MCA		
5	62	Atharv .V	Sem3	MCA		

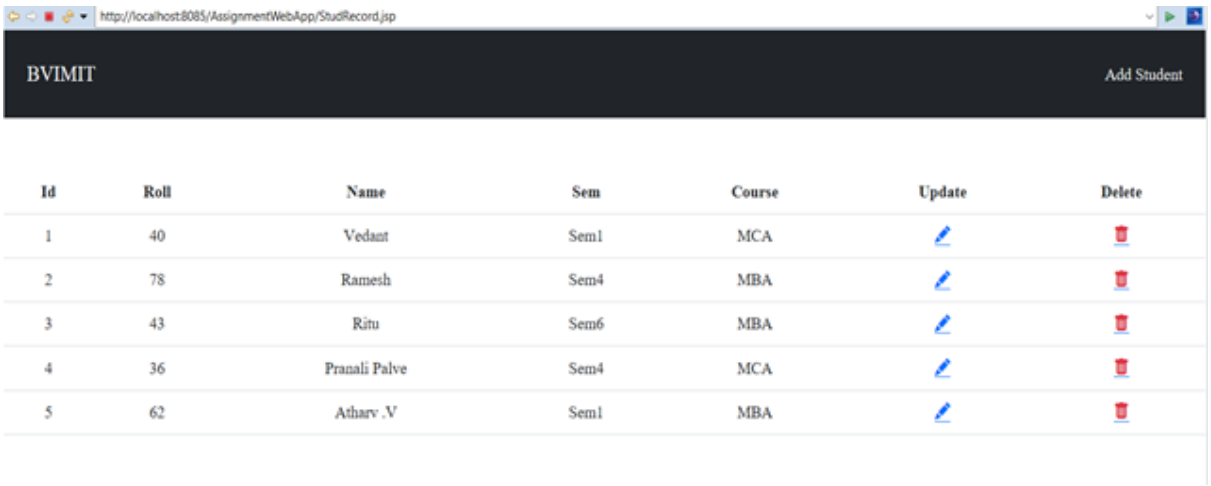
Update data-


http://localhost:8085/AssignmentWebApp/update.jsp?id=5
BVIMIT


62
Atharv .V
Semester 1
MBA
Update


Message from webpage
Updation Successful.
OK

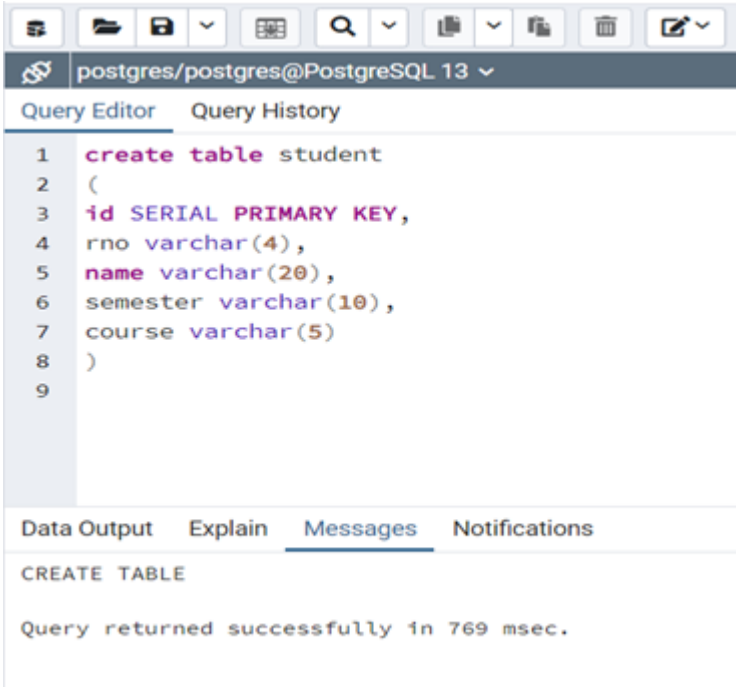

SQL Results
Execution Plan
Bookmarks
Console
Progress
Tomcat v8.5 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre1.8.0_202\bin\javaw.exe (28-Nov-2024, 4:13:
Opened database successfully
Opened database successfully
62
Atharv .V
Sem1
MBA
UPDATE student SET rno= ?, name = ?, semester = ?, course = ? WHERE id = '5'
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		
5	62	Atharv .V	Sem1	MBA		

Database:
CREATE :

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



Problem Statement 6.4 : Design loan calculator using JSP which accepts Period of Time (in years) and Principal Loan Amount

Solution:

Cal.jsp

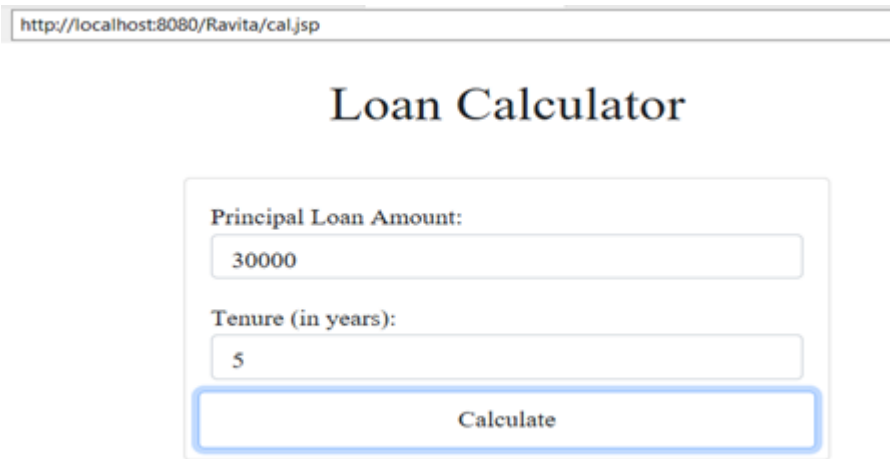
```
<%@ page language="java"contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html><head>
<meta charset="ISO-8859-1"><title>Load Calculator</title>
<!-- CSS only -->
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.4.0/font/bootstrap-icons.css">
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css"rel="styl
esheet"integrity="sha384-BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+
nYRRuWlolflfl"crossorigin="anonymous">
<!-- JavaScript Bundle with Popper -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/js/bootstrap.bundle.min.js"integr
ity="sha384-b5kHyXgcpbZJO/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S
9FOnJ0"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>
```

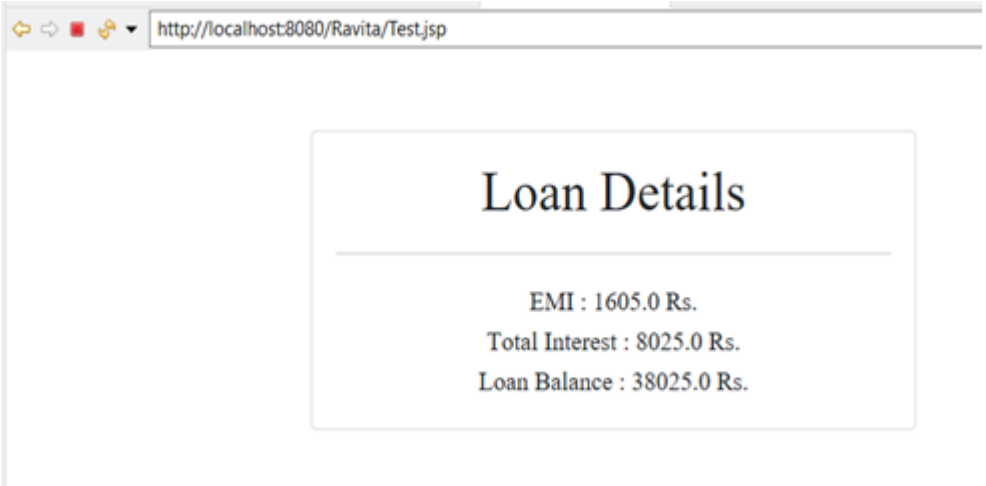
Text.jsp

```
<%@ 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="styl
esheet"integrity="sha384-BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+
nYRRuWlolflfl"crossorigin="anonymous">
<!-- JavaScript Bundle with Popper -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/js/bootstrap.bundle.min.js"integr
ity="sha384-b5kHyXgcpbZJO/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S
9FOnJ0"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-





Problem Statement 6.5 : 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.
Solution :

Filename-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>
```

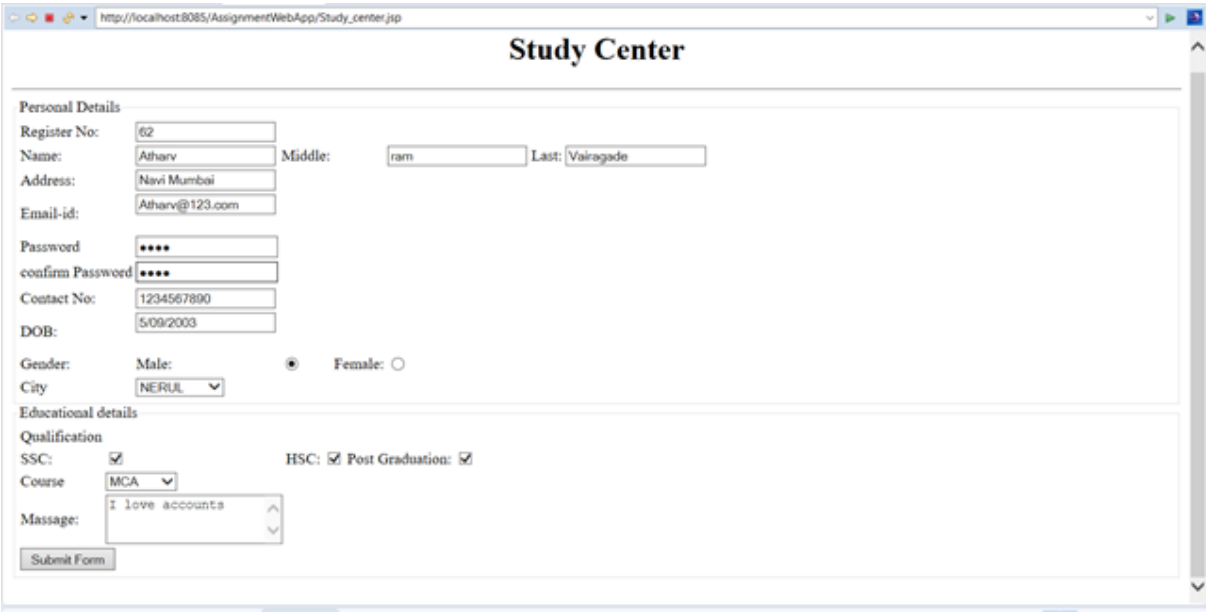
```
<option value="New Delhi">PANVEL</option>
<option value="Mumbai">KAMOTHE</option>
<option value="Goa">NERUL</option>
<option value="Patna">VASHI</option>
</select></td>
</tr>
</table>
</fieldset>
<fieldset>
<legend>Educational details</legend>
<table style="with: 50%">
<tr>
<td>Qualification</td>
</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

Problem Statement 6.6 : Write a JSP program that demonstrates the use of JSP declaration, scriptlet, directives, expression, header and footer.

Solution:

Filename-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>
```

Filename- header.jsp

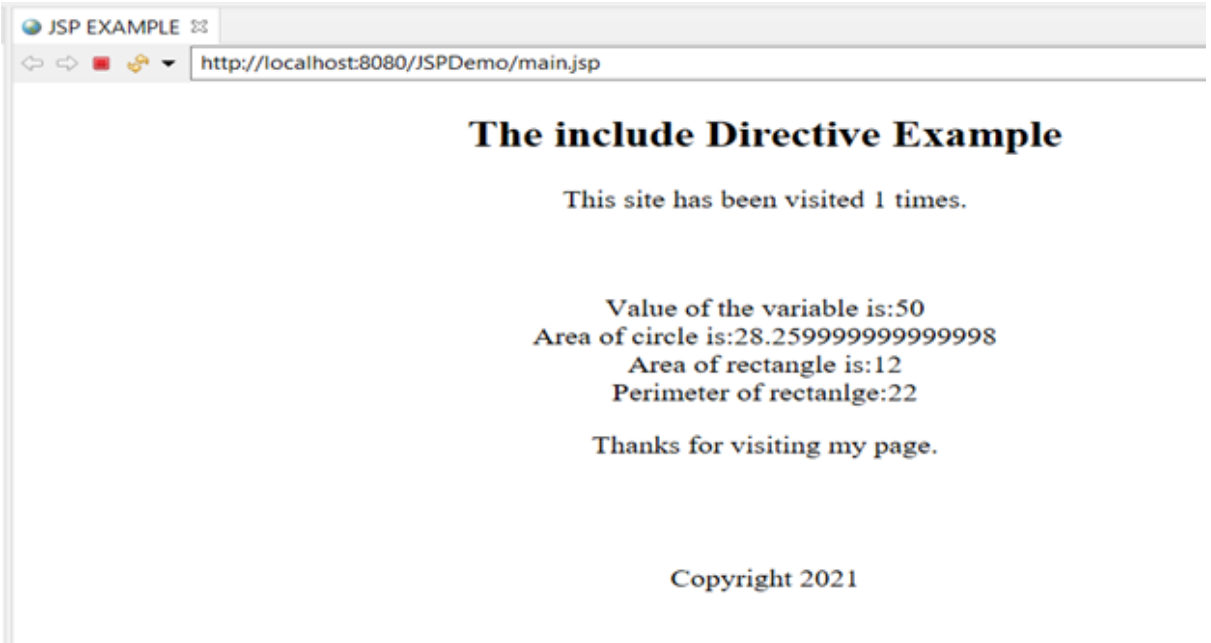
```
<%@ page language="java"contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<%!
 int pageCount = 0;
 void addCount() {
 pageCount++;
 }
%>
<% addCount(); %>

<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-


JSP EXAMPLE
http://localhost:8080/JSPDemo/main.jsp
The include Directive Example
This site has been visited 1 times.
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

6.7 Write a JSP program that demonstrates the use of session or cookies.

Solution:

```
<%@pagelanguage="java"contentType="text/html;charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPEhtml>
<html>
<head>
<metacharset="ISO-8859-1">
<title>Inserttitlehere</title>
</head>
<body>
<formaction="SessionCookie.jsp"method="post">
<table>
<tr>
<td>UserName:</td>
<td><inputtype="text"name="username"></td>
</tr>
<tr>
<td>Email:</td>
<td><inputtype="text"name="email"/></td></tr>
<tr>
<tdcolspan="2"><inputtype="submit"value="SubmitForm"/></td>
</tr>
</table>
</form>
<body>

<%
if(request.getParameter("username")!=null){
Cookie username = new Cookie("username",
request.getParameter("username"));Cookieemail=newCookie("email",request.getParameter("
email"));

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


//Addboththecookiesintheresponseheader.response.addCookie( username
);response.addCookie(email);

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

```
//GetanarrayofCookiesassociatedwiththethisdomaincookies=request.getCookies();

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



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

out.print(cookies[i].getValue()+"");
}
}else
{
out.println("<h2>Nocookiesfounds</h2>");
}
out.println(session.getAttribute("username"));out.println(session.getAttribute("email"))
}
%>

</body>
</html>
```

Output:

Spring Framework

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="Atharv"/>
```

```
</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 :

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="Atharv"></property>

<property name="age" value="21"></property>

</bean>

</beans>
```

SingerTest.java

```
package spring1;

import org.springframework.context.ApplicationContext;

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

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 :

```
Markers  Properties  Servers  Data Sour
<terminated> SingerTest [Java Application] C:\Progr
Name:Atharv Age:21
```

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="Belapur"></constructor-arg>

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

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

</bean>

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

<constructor-arg value="1" type="int"></constructor-arg>

<constructor-arg value="Atharv"></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;
import org.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:

```
Markers  Properties  Servers  Data Source E
<terminated> Test [Java Application] C:\Program Files\J
1 Atharv
Belapur mumbai India
```

Problem Statement 4 : Write a program to demonstrate Autowiring.

B.java

```
package
prac4;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/schem
    a/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">

<bean id="b" class="prac4.B"></bean>
<bean id="a" class="prac4.A" autowire="byName"></bean>
</beans>
```

Test.java

```
package org.bvimit;
import org.springframework.context.ApplicationContext;
import
org.springframework.context.support.ClassPathXmlApplicationConte
x t;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:-

```
Markers  Properties  Servers  Data Source Explorer
<terminated> Test (1) [Java Application] C:\Program Files\Java
b is created
a is created
hello a
hello b
```

Aspect Oriented Programming

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("Athrava B62");
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 (3) [AspectJ/Java Application] C:\java\New folder\bin\javaw.
Athrava B62
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("Athrava B62");
	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 (7) [AspectJ/Java Application] C:\java\New folder\
Athrava B62
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("Athrava B62");
	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 (9) [AspectJ/Java Application] C:\java\New folder\bin\javaw.ex
Athrava B62
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.AnnotationAwareAspectJAutoProxyCreat
or "></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("Athrava B62");

        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 (6) [AspectJ/Java Application] C:\java\New folder\bin\javaw.exe (28-Nov-2024,
Athrava B62
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.AnnotationAwareAspectJAutoProxyCreat
or ">
</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("Athrava B62");
	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:

```
Athrava B62
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
```

<terminated> TestValidation (7

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("Athrava B62");

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.AnnotationAwareAspectJAutoProxyCreator* "></bean>

</beans>

Output:

```
<terminated> Test_pc (1) [AspectJ/Java Application] C:\java\New fol
Athrava B62
calling m1...
method 1
calling m2...
method 2 with return
calling m3...
method 3 with return
```

Spring JDBC