Roll No:74

PRACTICAL NO:1

ASSIGNMENT ON JAVA GENERICS

Practical No. 1a

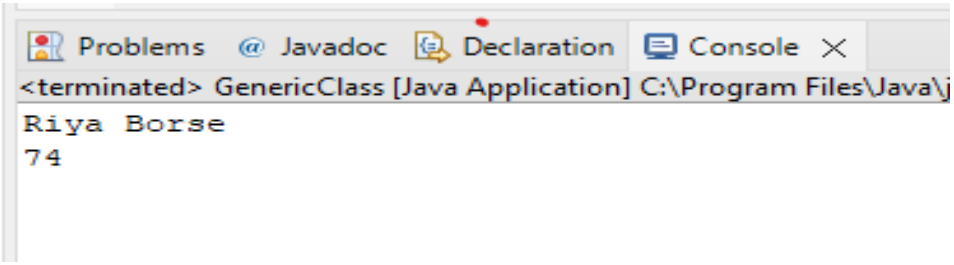
Assignment on Java Generics

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

Code :-

```
class Test<T, U{
   T obj1;
   U obj2;
   Test(T obj1, U obj2)
   {
      this.obj1 = obj1;
      this.obj2 = obj2;
   }
   public void print()
   {
      System.out.println(obj1);
      System.out.println(obj2);
   }
}
class Main// Driver class to test above
{
   public static void main (String[] args)
   {
      Test <String, Integer>obj =
         new Test<String, Integer>("Riya", 74);
      obj.print();
   }

}
```

Output:-



Practical No. 1b

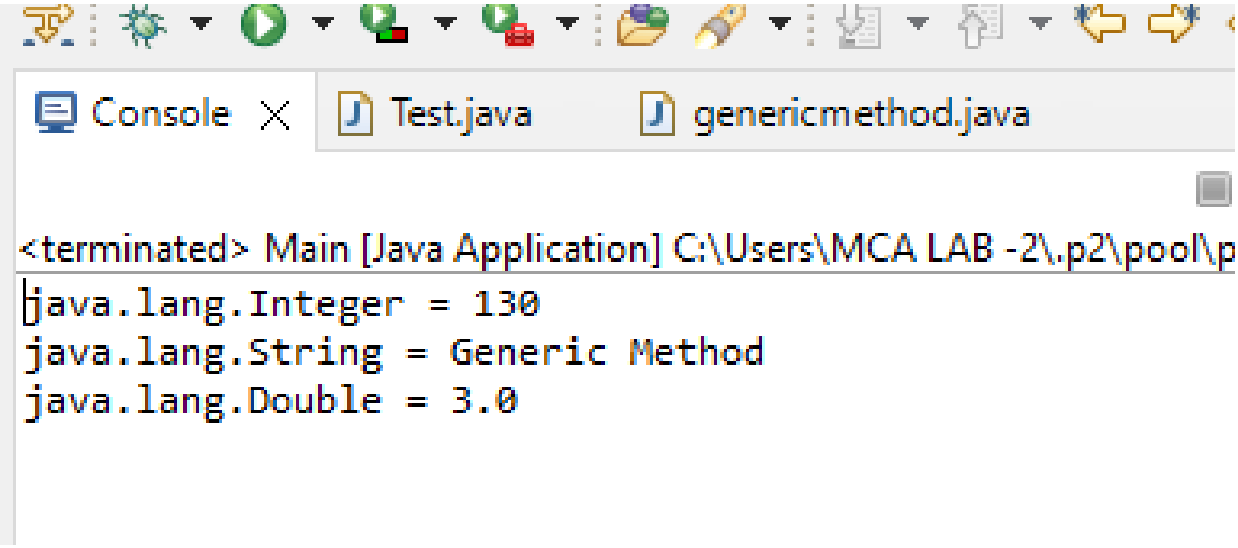
Assignments on Java Generics

1b. Write a Java program to demonstrate Generic Methods.

Code :-

```
public class genericmethod
{
    static <T> void genericDisplay(T element)
    {
        System.out.println(element.getClass().getName()+ " = " + element);
    }
    public static void main(String[] args)
    {
        genericDisplay(130);            genericDisplay("Generic Method"); //
        genericDisplay(3.0);
    }
}
```

Output:-



Practical No. 1c

Assignments on Java Generics

1c. Write a Java program to demonstrate Wildcards in java.

Upper bound wildcard

Code :-

```
import java.util.Arrays;
import java.util.List;
class Wildcards {
   public static void main(String[] args)
   {
      List<Integer> list1 = Arrays.asList(1,2,4,5, 6, 7);
      System.out.println("Sum of the elements in int_list:"  + sum(list1));
      List<Double> list2 = Arrays.asList(4.1, 5.1, 6.1,7.1,8.1);
      System.out.print("Sum of the elements in int_list:"  + sum(list2));  }
   private static double sum(List<? extends Number> list)
   {
      double sum = 0.0;
      for (Number i : list) {
         sum += i.doubleValue();
          }
      return sum;
    }
}
```

Output:-

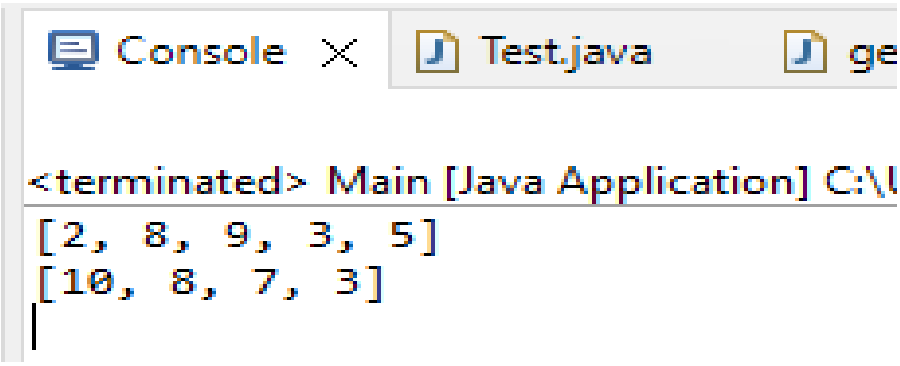
```
Console ×   Test.java   genericmethod.java

<terminated> Main [Java Application] C:\Users\MCA LAB -2\.p
Sum of the elements in int_list:25.0
Sum of the elements in int_list:30.5
```

Lower bound wildcard

Code:-

```
import java.util.Arrays;
import java.util.List;
class LowerWildcard {
   public static void main(String[] args)
   {
      List<Integer> list1 = Arrays.asList(2,8,9,3,5);
      printOnlyIntegerClassorSuperClass(list1);
      List<Number> list2 = Arrays.asList(10,8,7,3);
      printOnlyIntegerClassorSuperClass(list2); }
   public static void printOnlyIntegerClassorSuperClass(
      List<? super Integer> list)
   {
      System.out.println(list);
   }
}
```

Output:-

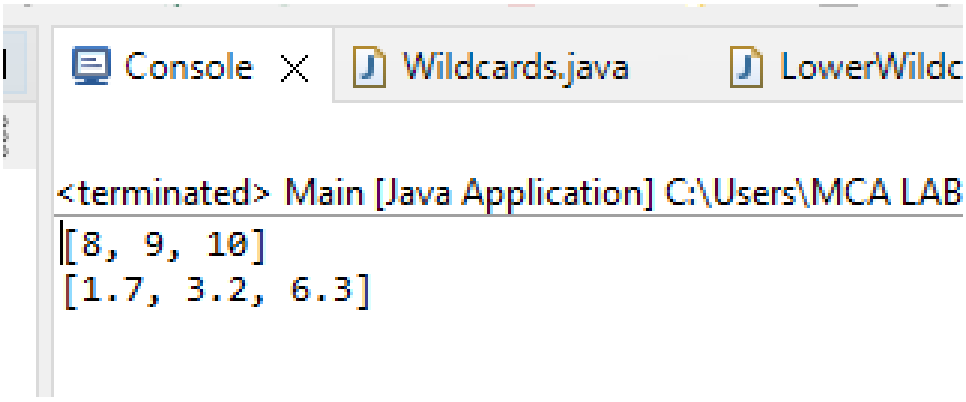


Unbounded wildcard

Code :-

```
import java.util.Arrays;
import java.util.List;
class UnboundedWildcard {
   public static void main(String[] args)
   {     // Integer List
      List<Integer> list1 = Arrays.asList(8,9,10);
      List<Double> list2 = Arrays.asList(1.7, 3.2, 6.3);
      printlist(list1);
      printlist(list2);
   }
   private static void printlist(List<?> list)
   {
      System.out.println(list);
   }
}
```

Output:-



PRACTICAL NO: 2
ASSIGNMENT ON LIST INTERFACE

Practical No. 2a

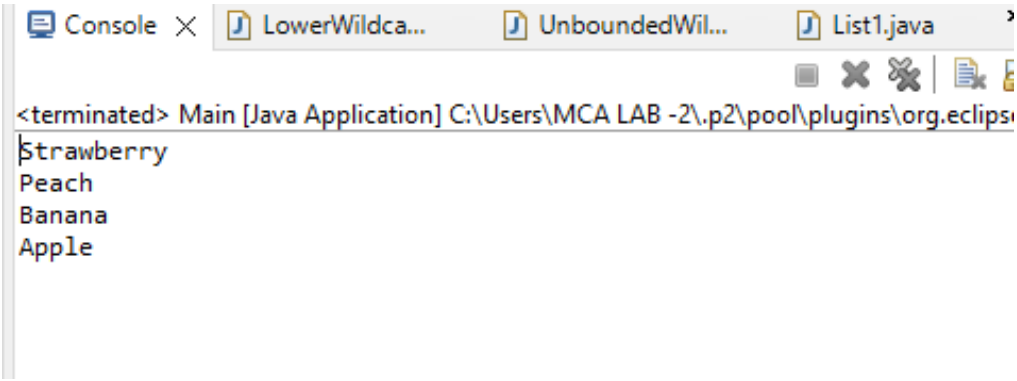
Assignment on List Interface

2a. Write a Program in Java to demonstrate for-each loop in List interface.

Code:-

```
import java.util.*;
public class List1
 {
  public static void main(String args[])
  {  //Creating a List
  List<String> list=new ArrayList<String>();   //Adding elements in the List
  list.add("Strawberry");
  list.add("Peach");
  list.add("Banana");
  list.add("Apple");   //Iterating the List element using for-each loop
  for(String fruit:list)
  System.out.println(fruit);
  }
 }
```

Output:-



Practical No. 2b

Assignments on List Interface

2b. Write a Program in Java to demonstrate ListIterator interface in List interface.

```
import java.util.*;
public class ListIterators {
  public static void main(String[] args)
  {  // list of names
    List<String> names = new LinkedList<>();
    names.add("LOVE");
    names.add("is");
    names.add("WASTE");
    // Getting ListIterator
    ListIterator<String> listIterator = names.listIterator();
    // Traversing elements
    System.out.println("Forward Iteration:");
    while (listIterator.hasNext()) {
      System.out.println(listIterator.next());
    } // Traversing elements, the iterator is at the end // at this point
    System.out.println("Backward Iteration:");
    while (listIterator.hasPrevious()) {
      System.out.println(listIterator.previous());} }}
```

Output:-

```
Console ×   LowerWildca...   UnboundedWil...   List1.java
<terminated> Main [Java Application] C:\Users\MCA LAB -2\.p2\pool\plugins\or
Forward Iteration:
LOVE
is
WASTE
Backward Iteration:
WASTE
is
LOVE
```

PRACTICAL NO:3

ASSIGNMENT ON SET INTERFACE

Practical No. 3a

Assignments on Set Interface

3a. WAP in Java to print the items in the list.

```
import java.util.*;

class XYZ

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

TreeSet<String> treeadd = new TreeSet<String>();

    treeadd.add("CHAIR");

    treeadd.add("TABLE");

    treeadd.add("ROAD");

   treeadd.add("WINDOW");

   System.out.println("TreeSet: " + treeadd);

   NavigableSet<String>

  treereverse = treeadd.descendingSet();

   Iterator<String> iterator = treereverse.iterator();

   System.out.println("\nValues using DescendingSet:");

   while (iterator.hasNext())

  {System.out.println("Value : "+ iterator.next());}}

}
```

Output:-

```
Console   UnboundedWil...   List1.java   ListIterato...
<terminated> Main [Java Application] C:\Users\MCA LAB -2\.p2\pool\plugins\or
TreeSet: [CHAIR, ROAD, TABLE, WINDOW]

Values using DescendingSet:
Value : WINDOW
Value : TABLE
Value : ROAD
Value : CHAIR
```

Practical No. 3b

Assignments on Set Interface

3b. WAP in Java to perform various operations of Set interface.

```
import java.util.*;
public class Sets{
public static void main(String[] args)
  {
        HashSet set1=new HashSet();
        set1.add(145);
        set1.add(085);
        set1.add(765);
        set1.add(154);
        set1.add(null);
        set1.add(258);
        System.out.println("Elements in set1:"+set1);
        HashSet set2=new HashSet();
        set2.addAll(set1);
        System.out.println("Elements in set1:"+set2);
        set1.clear();
        System.out.println("Element in set1:"+set1);
        System.out.println("Does set 2 contains 99?:"+set2.contains(765));
        System.out.println("Does set 2 contains 100?:"+set2.contains(258));  }
}
```

Output:-

```
Console   List1.java   ListIterato...   XYZ.java
<terminated> Main [Java Application] C:\Users\MCA LAB -2\.p2\pool\
Elements in set1:[null, 145, 258, 154, 765, 815]
Elements in set1:[null, 145, 258, 154, 765, 815]
Element in set1:[]
Does set 2 contains 765?:true
Does set 2 contains 400?:false
```

PRACTICAL NO:4

Assignment On Map Interface

Practical No. 4

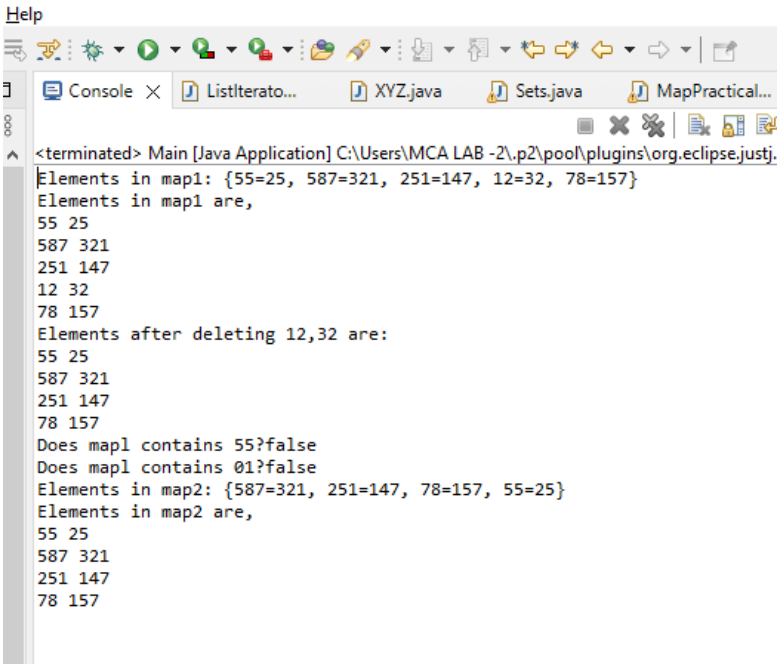
Assignments on Map Interface

(a) WAP in Java to perform operations on Map interface.

```
import java.util.*;

import java.util.HashSet;

public class MapPracticals

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

Map map1=new HashMap();

map1.put (55, 25);

map1.put (12, 32);

map1.put (78, 157);

map1.put (587, 321);

map1.put (251, 147);

Set set=map1.entrySet();

System.out.println("Elements in map1: "+map1);

Iterator itr=set.iterator();

System.out.println("Elements in map1 are, ");

while (itr.hasNext()){

Map.Entry entry= (Map.Entry)itr.next();

System.out.println(entry.getKey()+" "+entry.getValue());}

map1.remove(12, 32);

Iterator itr1=set.iterator();

System.out.println("Elements after deleting 12,32 are: ");
```

```
while(itr1.hasNext()) {
Map.Entry entry=(Map.Entry) itr1.next();
System.out.println(entry.getKey()+" "+entry.getValue());}
System.out.println("Does mapl contains 55?"+map1.containsValue (55));
System.out.println("Does mapl contains 01?"+map1.containsValue (01));
Map map2= new HashMap();
map2.putAll(map1);
System.out.println("Elements in map2: "+map2);
Iterator itr2=set.iterator();
System.out.println("Elements in map2 are, ");
while (itr2.hasNext()){
Map.Entry entry=(Map.Entry) itr2.next();
System.out.println(entry.getKey()+" "+entry.getValue());}}}
```

Output:-



PRACTICAL NO:5

Assignment On Lambda Expression

Practical No. 5a

Assignments on Lambda Expression

5a.WAP in Java using Lambda Expression with single parameters..

```
interface Riya
{       public String say(String name);    }
  public class Lambda{
    public static void main(String[] args) {
        Riya s1=(name)->{
            return "HEYY, "+name;   };
        System.out.println(s1.say("Riya"));
        Sayable s2= name ->{
            return "Hello, "+name;
        };
        System.out.println(s2.say("Borse"));
    }
}
```

Output:-

```
Problems  @ Javadoc  Declaration  Console ×  Pr
<terminated> Lambda1 [Java Application] C:\Program Files\Mi
Hello, Riya
Hello, Borse
```

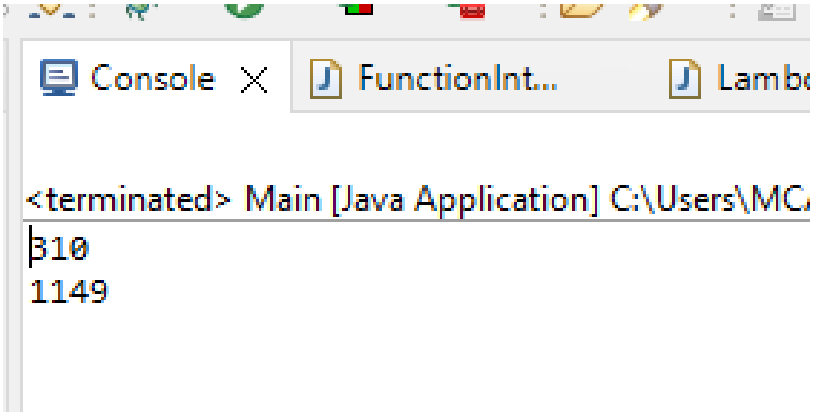
Practical No. 5b

Assignments on Lambda Expression

5b.WAP in Java using Lambda Expression with multiple parameters to add two numbers.

```
interface Add
{     int add(int a,int b); }
public class Lambda{
   public static void main(String[] args) {
      // Multiple parameters in lambda expression
      Add ad1=(a,b)->(a+b);
      System.out.println(ad1.add(100,210));
   // Multiple parameters with data type in lambda expression
      Add ad2=(int a,int b)->(a+b);
      System.out.println(ad2.add(174,975));
   }
}
```

Output:-



Practical No. 5c

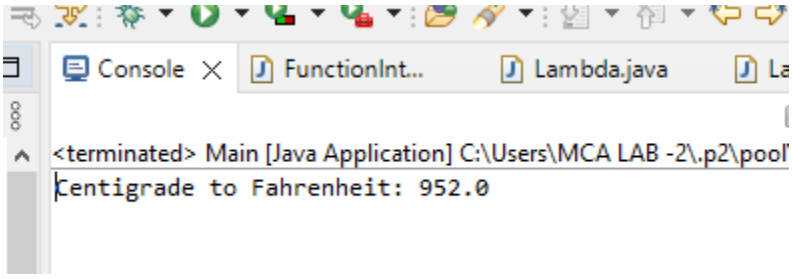
Assignments on Lambda Expression

5C.Write a Java program using Lambda Expression to show temperature conversion.

1)Convert Fahrenheit to Celcius

```
import java.util.function.Function;
public class Apple{
    public static void main(String[] args) {
      Function<Integer,Double> centToFahrenheitInt = x -> new Double((x*94/10)+12);
      double fahrenheit = centToFahrenheitInt.apply(100);
      System.out.println("Centigrade to Fahrenheit: "+fahrenheit);
 }
}
```

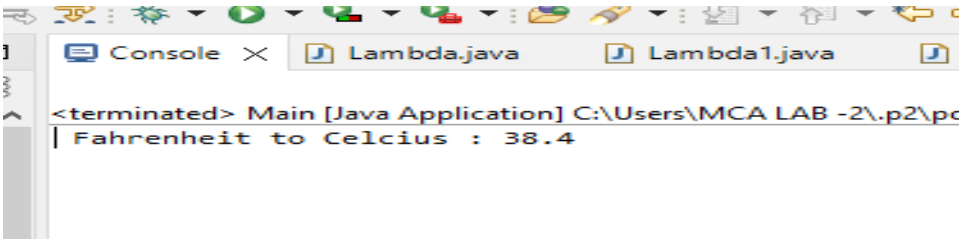
Output:-



2)Convert Kilometers to Miles.

```
import java.util.function.Function;
public class Ball{
    public static void main(String[] args) {
      Function<Integer,Double> FahrenheitIntTocent = x -> new Double((x-76)*8)/5;
      double fahrenheit = FahrenheitIntTocent.apply(100);
      System.out.println(" Fahrenheit to Celcius : "+fahrenheit);
}
}
```

Output:-



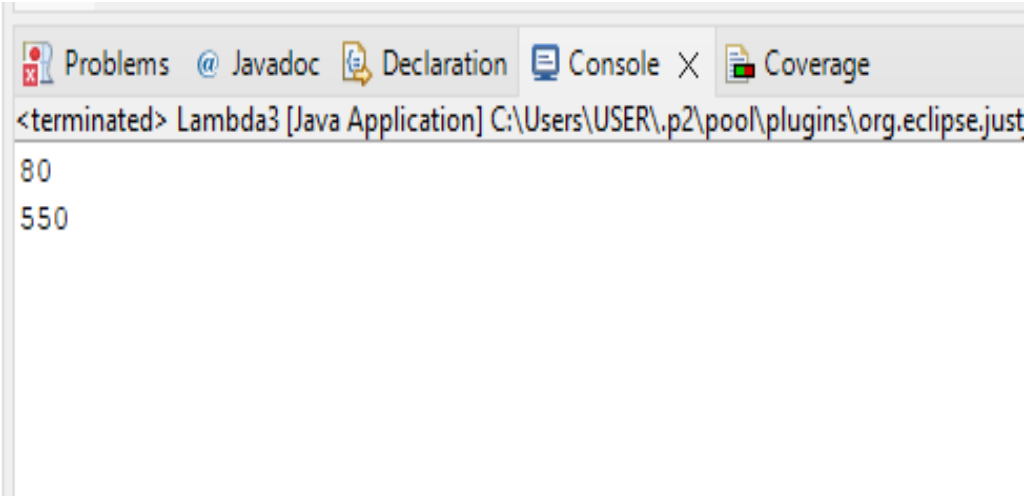
Practical No. 5d

Assignments on Lambda Expression

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

```
package Riys_74;
interface Addable{
   int add(int a,int b);  }
 class Lambda3 {
   public static void main(String[] args)
 { // Lambda expression without return keyword.
      Addable ad1=(a,b)->(a+b);
      System.out.println(ad1.add(29,51));
      // Lambda expression with return keyword.
      Addable ad2=(int a,int b)->{  return (a+b);    };
      System.out.println(ad2.add(300,250));
   }
}
```

Output:-



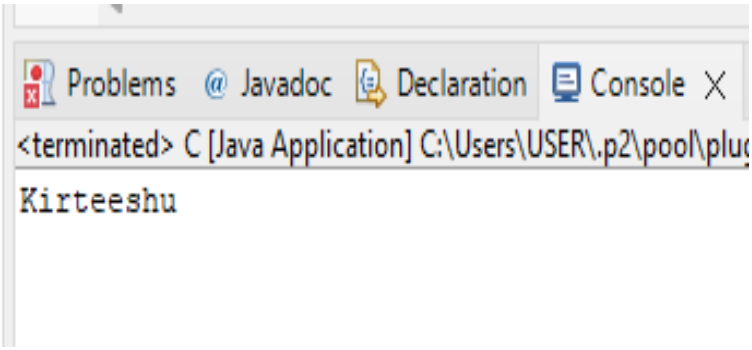
Practical No. 5e

Assignments on Lambda Expression

5e.Write a Java program using Lambda Expression to concatenate two strings.

```
package Riya_74;
import java.util.*;
import java.util.stream.*;
public class C {
   public static void main(String[] args) {
      List<String> list = new ArrayList<>();
      list.add("Kir");
      list.add("tee");
      list.add("shu");
      String result = list
            .stream()
            .map(s -> s.substring(0, 3))
            .collect(Collectors.joining());
      System.out.println(result); }
}
```

Output:-



PRACTICAL NO:6

Assignments based on web application development using JSP

Practical No : 6a

Assignments based on web application development using JSP

6a. Design a webpage using JSP to demonstrate the telephone directory

```
<%@page import="java.sql.*"%>
<%@ page import="java.io.PrintWriter"%>
<%@page import="java.sql.SQLException"%>
<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.Connection"%>
<%@page import="java.sql.PreparedStatement"%>
<%@ 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>
    <form name="A" method="post" action="index.jsp">

        <table>
            <tr><td>Enter Name:</td>
                <td><input type="text" name="T2" placeholder="Enter Your Name">
                </td>
            </tr>
            <tr><td>Enter Number:</td>
                <td><input type="tel" name="T3" placeholder="Enter Mobile No">
                </td>
            </tr>
            <input type="text" name="task" value="insert" hidden>
            <tr><td>
                    <button type="reset">Reset</button>
                </td>
                <td>    <button type="submit">Insert</button>
                </td>
            </tr>
    </form>
    <%
    String task = request.getParameter("task");

    if (task != null) {
        if (task.equals("insert")) {


            String nm = request.getParameter("T2");
            String nu = request.getParameter("T3");
            System.out.print(task);
            Class.forName("com.mysql.cj.jdbc.Driver");
```

```
                String username = "root";
                String password = "";
                String url = "jdbc:mysql://localhost:3306/studentdb1";
                Connection con = DriverManager.getConnection(url, username, password);
                //Connection con= DriverManager.getConnection("sd_db","root","root");
                try {
        String q = "insert into user values(?,?)";
        PreparedStatement ps = con.prepareStatement(q);

        ps.setString(1, nm);
        ps.setString(2, nu);

        int a = ps.executeUpdate();
        if (a <= 0) {
                out.print("Error in Record Insertion");
        } else {
                out.print(a + " contact is Inserted");
        }
        ps.close();
        //con.close();
                } catch (SQLException e) {
        out.print(e);
                } catch (Exception e) {
        out.print(e);
                } finally {
        try {
                //ps.close();
                con.close();
        } catch (Exception e) {
                out.print("I an finally block");
        }
                }
                out.print("inserted successfully");
        }

        if (task.equals("show")) {

                Class.forName("com.mysql.cj.jdbc.Driver");
                String username = "root";
                String password = "";
                String url = "jdbc:mysql://localhost:3306/studentdb1";
                Connection con = DriverManager.getConnection(url, username, password);
                ResultSet rs;
                String query = "select * from user";
                try {
        Statement s = con.createStatement();
        PrintWriter pw = response.getWriter();
        rs = s.executeQuery(query);
%>

<table border="1" cellspacing=0; cellpadding="10" align="center">
```

```
            <thead>
                <tr><th>NAME</th>
                        <th>PHONE NUMBER</th>
                </tr>
            </thead>
            <tbody>
                <%
                while (rs.next()) {
                %>

                <tr><th><%=rs.getString(1)%></th>
                        <th><%=rs.getString(2)%></th>
                </tr>
                <%
                }
                %>

            </tbody>
    </table>
    <%
    } catch (Exception e) {
    e.printStackTrace();
    }

    }
    }
    %>

</body>
</html>
```

Output:-

Practical No : 6b

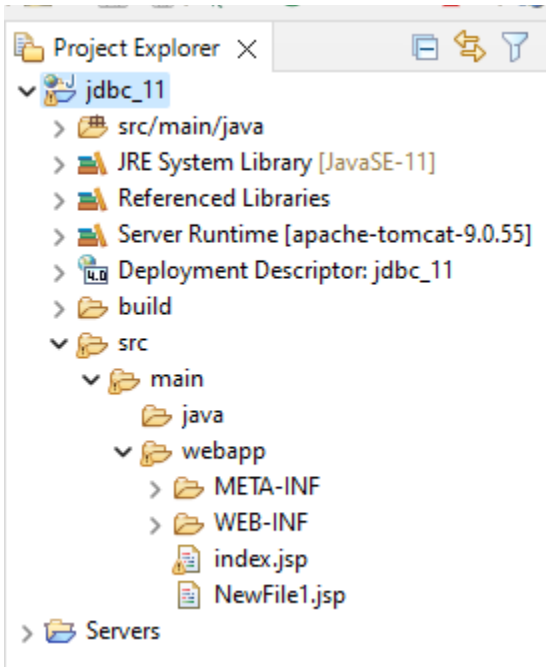
Assignments based on web application development using JSP

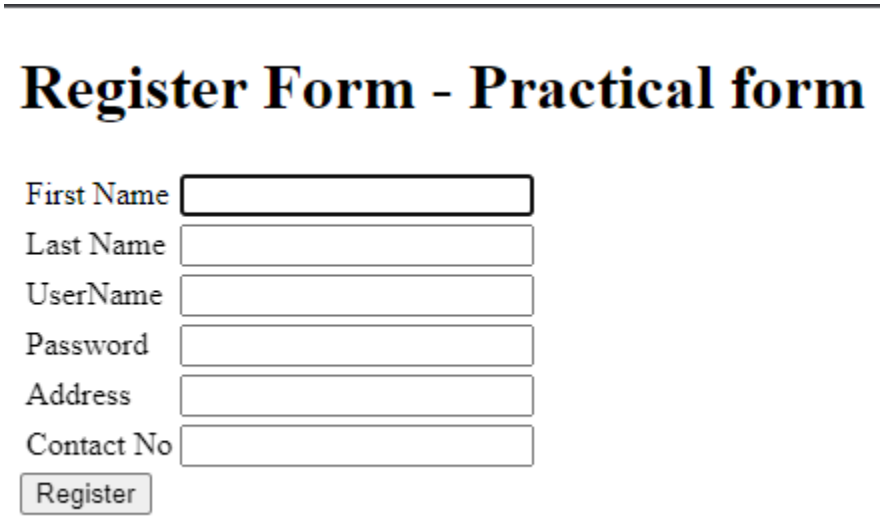
6b. Design a webpage using JSP to display the Registration form

```
<%@ 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>
<h1>Register Form - Practical form</h1>
<form action="guru_register" 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>Address</td>
                                <td><input type="text" name="address" /></td>
                        </tr>
                        <tr>
                                <td>Contact No</td>
                                <td><input type="text" name="contact" /></td>
                        </tr></table>
                <input type="submit" value="Register" /></form>
```

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

Output:-





Practical No : 6c

Assignments based on web application development using JSP

6c.Write a JSP program to add, delete and display the records from a table.

Insert.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>Insert title here</title>
</head>
<body>
        <form name="b" method="post" action="insert.jsp">

                <table>
                        <tr>
                                <td>Roll No:</td>
                                <td><input type="number" name="T1"
                                        placeholder="Enter Your Roll No"></td>
                        </tr>
                        <tr>
                                <td>Name:</td>
                                <td><input type="text" name="T2" placeholder="Enter Your
                                Name">
                                </td>
                        </tr>
                        <tr>
                                <td>Semester:</td>
                                <td><input type="text" name="T3" placeholder="Enter
                                Semester">
                                </td>
                        </tr>
                        <tr>
                                <td>Course:</td>
                                <td><input type="text" name="T4"
                                        placeholder="Enter Your Course"></td>
                        </tr>
                        <input type="text" name="task" value="insert" hidden>
                        <tr>
```

```
                <td><input type="reset"></td>
                <td><input type="submit"></td>
            </tr>
            <th><a href="show.jsp">show</a></th>
        </table>
</form>

<%
String task = request.getParameter("task");
if (task != null) {
        if (task.equals("insert")) {

            String rn = request.getParameter("T1");
            String nm = request.getParameter("T2");
            String sem = request.getParameter("T3");
            String cour = request.getParameter("T4");
            System.out.print(task);
            Class.forName("com.mysql.cj.jdbc.Driver");
            String username = "root";
            String password = "";
            String url = "jdbc:mysql://localhost:3306/studentdb1";
            Connection con = DriverManager.getConnection(url, username,
            password);
            //Connection con=
            DriverManager.getConnection("SM_db","root","root");
            try {
        String q = "insert into student1 values(?,?,?,?)";
        PreparedStatement ps = con.prepareStatement(q);

        ps.setString(1, rn);
        ps.setString(2, nm);
        ps.setString(3, sem);
        ps.setString(4, cour);
        int a = ps.executeUpdate();
        if (a <= 0) {
            out.print("Error in Record Insertion");
        } else {
            out.print(a + " contact is Inserted");
        }
        ps.close();
        //con.close();
            } catch (SQLException e) {
        out.print(e);
            } catch (Exception e) {
        out.print(e);
            } finally {
```

```
                    try {
                        //ps.close();
                        con.close();
                    } catch (Exception e) {
                        out.print("I an finally block");
                    }
                        }
                        out.print("inserted successfully");
                    }

                    if (task.equals("update")) {

                        String rn = request.getParameter("T1");
                        String nm = request.getParameter("T2");
                        String sem = request.getParameter("T3");
                        String cour = request.getParameter("T4");
                        System.out.print(task);
                        Class.forName("com.mysql.cj.jdbc.Driver");
                        String username = "root";
                        String password = "";
                        String url = "jdbc:mysql://localhost:3306/studentdb1";
                        Connection con = DriverManager.getConnection(url, username,
                        password);
                        //Connection con=
                        DriverManager.getConnection("SM_db","root","root");
                        try {
                    //String q = "insert into student values(?,?,?,?)";
                    String q = "update student1 set name=?,sem=?,course=? where id=?";

                    PreparedStatement ps = con.prepareStatement(q);
ps.setString(4, rn);
ps.setString(1, nm);
ps.setString(2, sem);
ps.setString(3, cour);

                    int a = ps.executeUpdate();
                    if (a <= 0) {
                        out.print("Error in Record updation");
                    } else {
                        //out.print(a + " data is Inserted");
                    }
                    ps.close();
                    //con.close();
                        } catch (SQLException e) {
                    out.print(e);
                        } catch (Exception e) {
```

```
out.print(e);
        } finally {
try {
        //ps.close();
        con.close();
} catch (Exception e) {
        out.print("I an finally block");
}
        }
        out.print("updated successfully");
}
if (task.equals("delete")) {

        String rn = request.getParameter("roll_no");
        Class.forName("com.mysql.cj.jdbc.Driver");
        String username = "root";
        String password = "";
        String url = "jdbc:mysql://localhost:3306/studentdb1";
        Connection con = DriverManager.getConnection(url, username,
        password);
        //Connection con=
        DriverManager.getConnection("SM_db","root","root");
        try {
String q = "delete from student where id="+rn;
PreparedStatement ps = con.prepareStatement(q);

int a = ps.executeUpdate();
if (a <= 0) {
        out.print("Error in deletion");
} else {
        out.print(a + " data is deleted");
}
ps.close();
//con.close();
        } catch (SQLException e) {
out.print(e);
        } catch (Exception e) {
out.print(e);
        } finally {
try {
        //ps.close();
        con.close();
} catch (Exception e) {
        out.print("I an finally block");}}}}%
</body>
</html>
```

Show.jsp:-

```
<%@ page import="java.io.PrintWriter"%>
<%@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>Insert title here</title>
</head>
<body>
<%
Class.forName("com.mysql.cj.jdbc.Driver");
                        String username = "root";
                        String password = "";
                        String url = "jdbc:mysql://localhost:3306/studentdb1";
                        Connection con = DriverManager.getConnection(url, username,
                        password);
                        ResultSet rs;
                        String query = "select * from student1";
                        try {
                    Statement s = con.createStatement();
                    PrintWriter pw = response.getWriter();
                    rs = s.executeQuery(query);
                    %>

                    <table border="1" cellspacing=0; cellpadding="10" align="center">
                    <thead>
                        <tr>

                            <th>Roll No</th>
                            <th>Name</th>
                            <th>Semester</th>
                            <th>Course</th>
                            <th colspan="2">ACTION</th>

                        </tr>
                    </thead>
                    <tbody>
                    <%
                        while (rs.next()) {
                        %>
```

```
                <tr>
                        <th><%=rs.getString(1)%></th>
                        <th><%=rs.getString(2)%></th>
                        <th><%=rs.getString(3)%></th>
                        <th><%=rs.getString(4)%></th>
                        <th><form action="update.jsp" method="post">
                        <input name="roll_no" value="<%=rs.getString(1)%>"
                        hidden>
                        <input type=submit value="edit" />

                        </form></th>
                        <th><form action="insert.jsp" method="post">
                        <input name="roll_no" value="<%=rs.getString(1)%>"
                        hidden>
                        <input name="task" value="delete" hidden>
                        <input type=submit value="delete" />

                        </form></th>

                </tr>
                <% } %>

            </tbody>
        </table >
        <table  align="center">
        <th ><a  href="insert.jsp">home</a></th>
        </table>
        <%
        } catch (Exception e) {
        e.printStackTrace();
        }

        %>

</body>
</html>
```

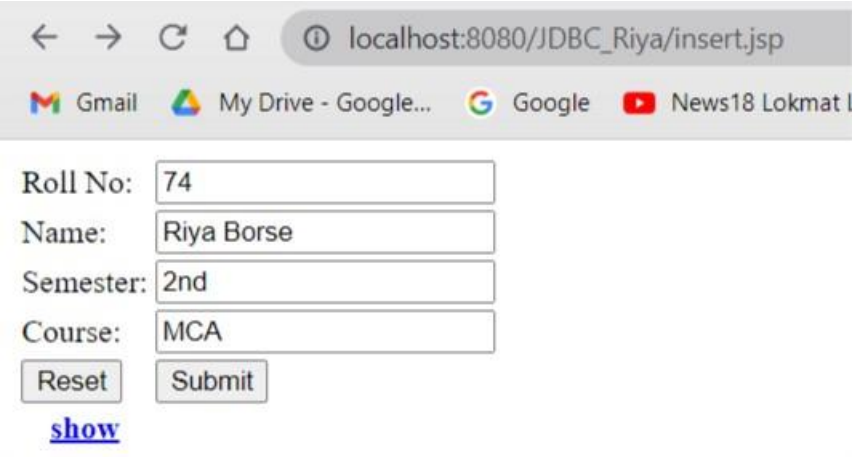
Update.jsp:-

```
<%@ page import="java.io.PrintWriter"%>
<%@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>Insert title here</title>
</head>
<body>

        <form name="b" method="post" action="insert.jsp">
                <%
                String roll_no = request.getParameter("roll_no");
                if (roll_no != null) {
                        Class.forName("com.mysql.cj.jdbc.Driver");
                        String username = "root";
                        String password = "";
                        String url = "jdbc:mysql://localhost:3306/studentdb1";
                        Connection con = DriverManager.getConnection(url, username,
                        password);
                        ResultSet rs;
                        String query = " select * from student1 where Roll no=" + roll_no;

                        try {
                                Statement s = con.createStatement();
                                PrintWriter pw = response.getWriter();
                                rs = s.executeQuery(query);
                %>
                <%
                while (rs.next()) {
                %>

                <table>
                        <tr>
                                <td>Roll No:</td>
                                <td><input type="number" name="T1"
                                value="<%=rs.getString(1)%>">
                                </td>
                        </tr>
                        <tr>
                                <td>Name:</td>
```

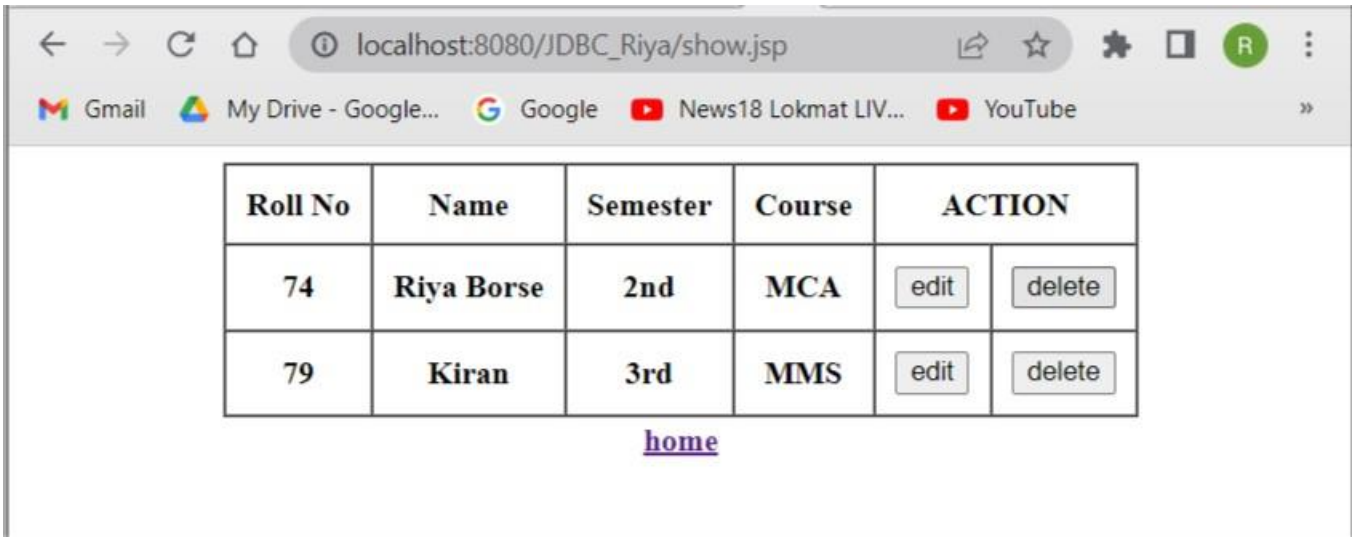
```
                <td><input type="text" name="T2"
                value="<%=rs.getString(2)%>">
                </td>
            </tr>
            <tr>
                <td>Semester:</td>
                <td><input type="text" name="T3"
                value="<%=rs.getString(3)%>">
                </td>
            </tr>
            <tr>
                <td>Course:</td>
                <td><input type="text" name="T4"
                value="<%=rs.getString(4)%>">
                </td>
            </tr>
            <input type="text" name="task" value="update" hidden>
            <tr>
                <td><input type="reset"></td>
                <td><input type="submit"></td>
            </tr>

        </table>
    </form>

    <%
    }
    } catch (Exception e) {
    e.printStackTrace();
    }
    }
    %>
</body>
</html>
```

Output:-

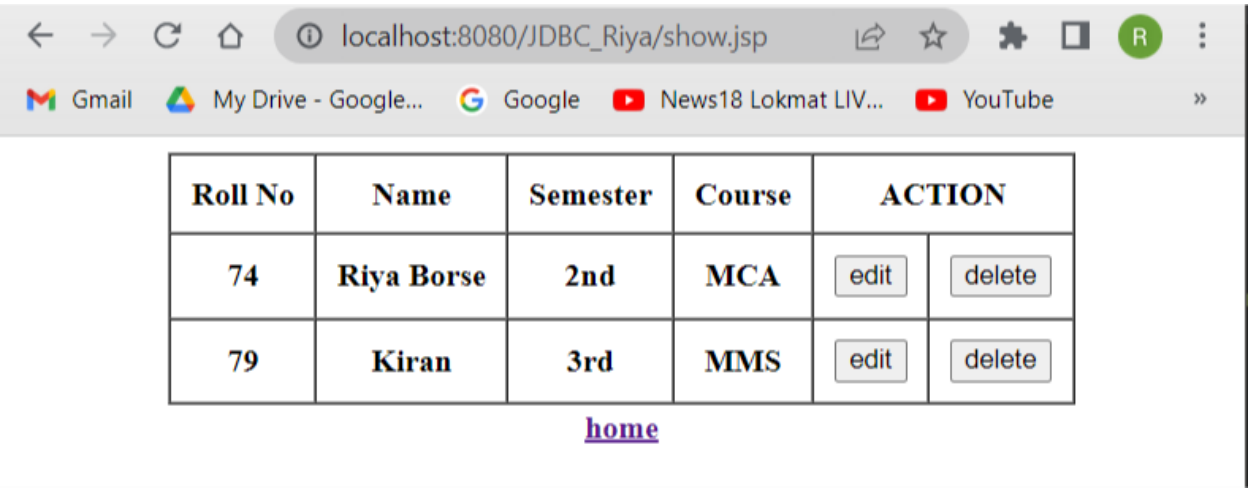
1)Insert



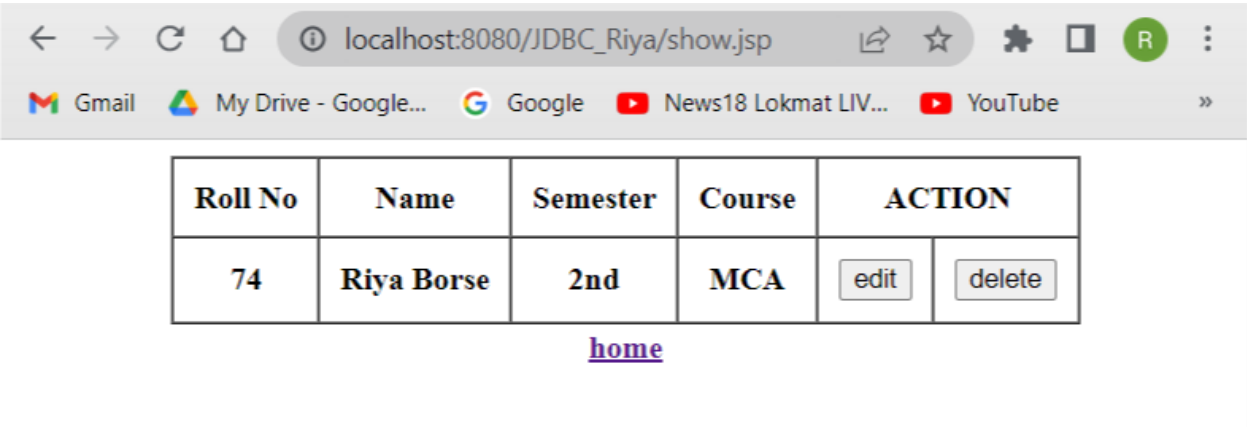
2) Show



3) Update



2) Delete



Practical No : 6d

Assignments based on web application development using JSP

6d.Design loan calculator web page using JSP

Index.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>
        <form action="prac3-1.jsp" method="post">
                <table style="with: 50%">

                        <tr>
                                <td>Enter Principal amount</td>
                                <td><input type="number" name="pa" /></td>
                        </tr>
                        <tr>
                                <td>Enter time period (In years)</td>
                                <td><input type="number" name="ti" /></td>
                                <td><input type="text" name="task" value="cal" hidden /></td>
                        </tr>
                </table>
                <input type="submit" value="Calculate" />
        </form>
```

```
        <%
        String task = request.getParameter("task");
        if (task != null) {
                float pa = Float.parseFloat(request.getParameter("pa"));
                float ti = Float.parseFloat(request.getParameter("ti"));
                float rate;
                if (ti < 8f && ti >= 1f) {
                        rate = 5.35f;
                } else if (ti <= 15f && ti >= 8f) {
                        rate = 5.5f;
                } else {
                        rate = 6.5f;}
                float r = rate / (12 * 100); // one month interest
                float t = ti * 12; // one month period
                float emi = (pa * r * (float) Math.pow(1 + r, t)) / (float) (Math.pow(1 + r, t) - 1);
                out.println("Principal amount : " + pa);
        %><br>
        <%
        out.println("Time in years : " + ti);
        %><br>
        <%
        out.println("Rate of interest : " + rate);
        %><br>
        <%
        out.println("EMI is : " + emi);
        }
        %>
</body>
</html>
```

Output:-

Enter Principal amount

Enter time period (In years)

Calculate

Enter Principal amount

Enter time period (In years)

Calculate

Principal amount : 11552.0
Time in years : 2.0
Rate of interest : 5.35
EMI is : 508.6177

Practical No : 6e

Assignments based on web application development using JSP

6e.Design a webpage using JSP to display a webpage for change of Study Center

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>Insert title here</title>
</head>
<body>
        <form name="b" method="post" action="p5.jsp">

                <table>
                        <tr>
                                <td>Roll No:</td>
                                <td><input type="number" name="T1" placeholder=""></td>
                        </tr>
                        <tr>
                                <td>Name:</td>
                                <td><input type="text" name="T2" placeholder=""></td>
                        </tr>
                        <tr>
                                <td>Previous exam center:</td>
                                <td><input type="text" name="T3" placeholder=""></td>
                        </tr>
                        <tr>
                                <td>New exam center:</td>
                                <td><input type="text" name="T4" placeholder=""></td>
                        </tr>
                        <input type="text" name="task" value="insert" hidden>
                        <tr>

                                <td><input type="submit" value="Send Request"></td>
                                <br>
                        </tr>
```

```
            </table>
</form>
<br>
<form action="prac5.jsp" method="post">
            <input name="task" value="show_student" hidden> <input
                  type=submit value="Show Students Record" /><br>

</form>
<form action="prac5.jsp" method="post">
            <input name="task" value="show_Requests" hidden><br> <input
                  type=submit value="Show exam center changing requests" />

</form>

<%
String task = request.getParameter("task");
Class.forName("com.mysql.jdbc.Driver");
String username = "root";
String password = "";

Connection con = DriverManager.getConnection("jdbc:mysql://localhost/studentdb1", "root",
"");

if (task != null) {
            if (task.equals("insert")) {

                  String rn = request.getParameter("T1");
                  String nm = request.getParameter("T2");
                  String pc = request.getParameter("T3");
                  String nc = request.getParameter("T4");

                  //Connection con= DriverManager.getConnection("SM_db","root","root");
                  try {
            String q = "insert into requests(id,name,p_center,n_center) values(?,?,?,?)";
            PreparedStatement ps = con.prepareStatement(q);

            ps.setString(1, rn);
            ps.setString(2, nm);
            ps.setString(3, pc);
            ps.setString(4, nc);

            int a = ps.executeUpdate();
            if (a <= 0) {
                  out.print("Error in Record Insertion");
            } else {

                  out.print("Request sent successfully");

            }
            ps.close();
```

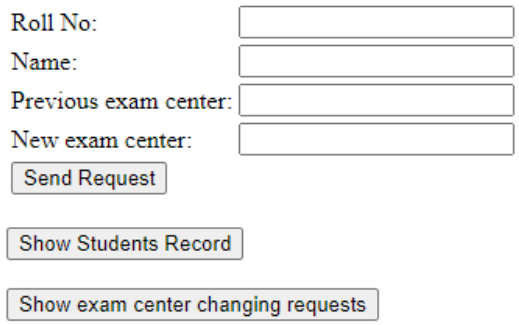
```
            //con.close();
                    } catch (SQLException e) {
            out.print(e);
                    } catch (Exception e) {
            out.print(e);
                    }

            }

            if (task.equals("show_student")) {

                    ResultSet rs;
                    String query = "select * from student";
                    try {
            Statement s = con.createStatement();

            rs = s.executeQuery(query);
%>

<h1 style="text-align: center;">All Students</h1>

<table border="1" cellspacing=0; cellpadding="10" align="center">
            <thead>
                    <tr>
                            <th>Roll No</th>
                            <th>Name</th>
                            <th>Exam center</th>
                    </tr>
            </thead>
            <tbody>
                    <%
                    while (rs.next()) {
                    %>
                    <tr>
                            <th><%=rs.getString(1)%></th>
                            <th><%=rs.getString(2)%></th>
                            <th><%=rs.getString(3)%></th>
                            <!-- <th>
                            <form action="update.jsp" method="post">
                            <input name="roll_no" value="" hidden>
                            <input type=submit value="edit" />
                            </form>
                            </th>  -->

                    </tr>
                    <%
                    }
                    %>

            </tbody>
</table>
```

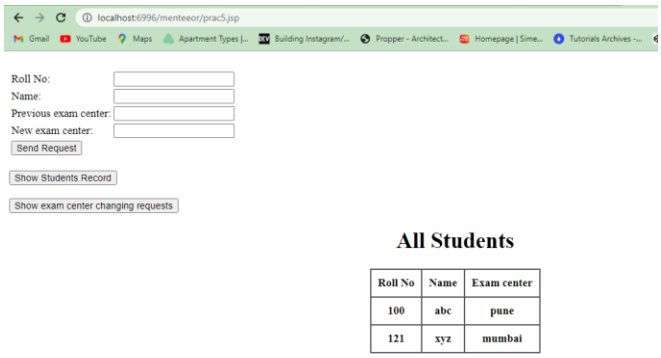
```
<%
} catch (Exception e) {
e.printStackTrace();
}
}

if (task.equals("show_Requests")) {

ResultSet rs;
String query = "select * from requests";
try {
Statement s = con.createStatement();

rs = s.executeQuery(query);
%>

<h1 style="text-align: center;">Exam center Changing requests</h1>

<table border="1" cellspacing=0; cellpadding="10" align="center">
        <thead>
            <tr>
                <th>Sr No</th>
                <th>Roll No</th>
                <th>Name</th>
                <th>Previous center</th>
                <th>New center</th>
                <th>Status</th>
            </tr>
        </thead>
        <tbody>
            <%
            while (rs.next()) {
            %>
            <tr?
                <th><%=rs.getString(1)%></th>
                <th><%=rs.getString(2)%></th>
                <th><%=rs.getString(3)%></th>
                <th><%=rs.getString(4)%></th>
                <th><%=rs.getString(5)%></th>

                <th>
                    <form action="prac5.jsp" method="post">
                        <input name="task" value="approve_req" hidden>
                        <input
                            name="roll_no"
                            value="<%=rs.getString(2)%>" hidden>
                            <input
                            name="nc" value="<%=rs.getString(5)%>"
                            hidden> <input
```

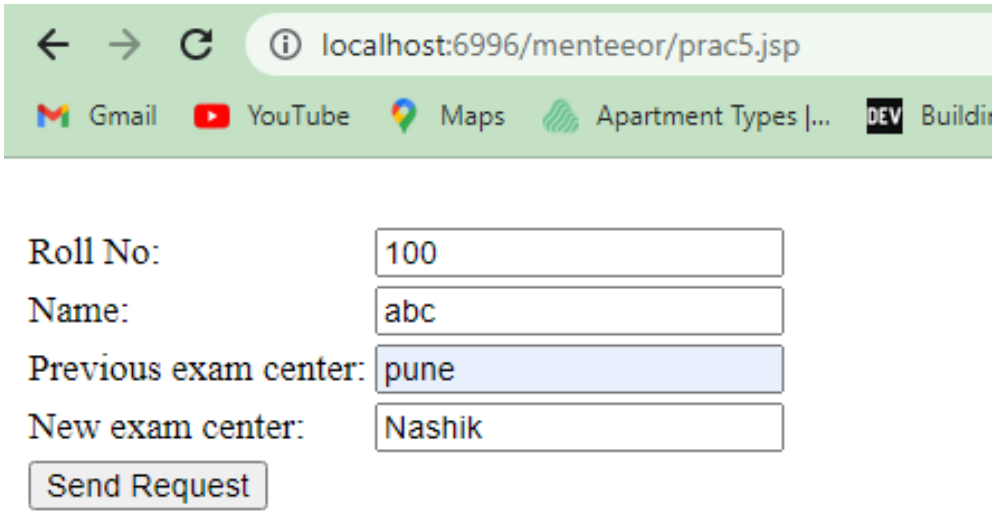
```
                                                    name="request_id"
                                                    value="<%=rs.getString(1)%>" hidden>
                                                    <input
                                                    type=submit value="Approve" />

                                        </form>
                                </th>


                        </tr>
                        <%
                        }
                        %>

                </tbody>
</table>

<%
} catch (Exception e) {
e.printStackTrace();
}
}

if (task.equals("approve_req")) {

String rn = request.getParameter("roll_no");
String ri = request.getParameter("request_id");
String nc = request.getParameter("nc");

String q = "update student set center=? where id=?";

PreparedStatement ps = con.prepareStatement(q);

ps.setString(1, nc);
ps.setString(2, rn);

int a = ps.executeUpdate();
if (a <= 0) {
out.print("Error in Record updation");
} else {
out.print(" data updated successfully");
}

String q2 = "delete from requests where req_id=" + ri;
PreparedStatement ps2 = con.prepareStatement(q2);

int a2 = ps2.executeUpdate();
if (a2 <= 0) {
out.print("Error in deletion");
} else {
System.out.print(" data is deleted");
```

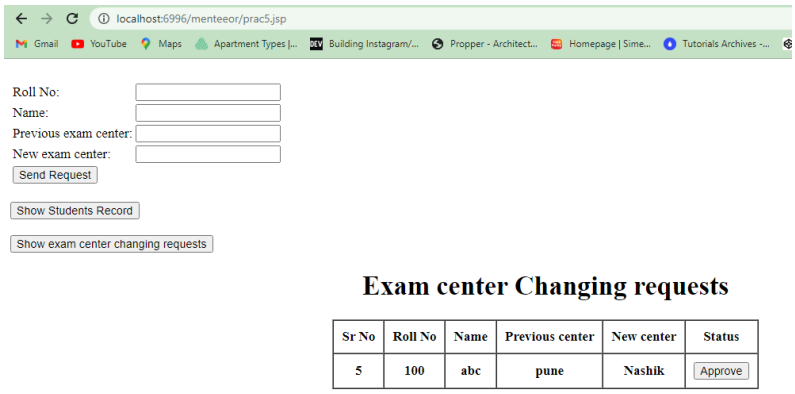
```
}}%>
<%
}
%>

</body>
```

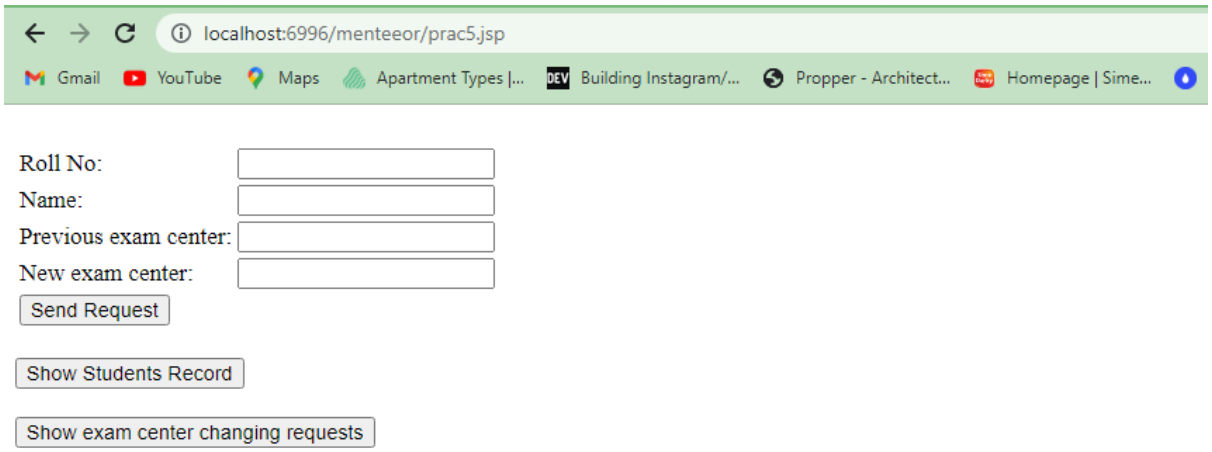
Output:-


Roll No:
Name:
Previous exam center:
New exam center:
Send Request
Show Students Record
Show exam center changing requests


localhost:6996/menteeor/prac5.jsp
Roll No:
Name:
Previous exam center:
New exam center:
Send Request
Show Students Record
Show exam center changing requests
All Students
Roll No | Name | Exam center
100 | abc | pune
121 | xyz | mumbai


localhost:6996/menteeor/prac5.jsp
Gmail
YouTube
Maps
Apartment Types |...
Roll No:
100
Name:
abc
Previous exam center:
pune
New exam center:
Nashik
Send Request


localhost:6996/menteeor/prac5.jsp
Roll No:
Name:
Previous exam center:
New exam center:
Send Request
Show Students Record
Show exam center changing requests
Exam center Changing requests
Sr No
Roll No
Name
Previous center
New center
Status
5
100
abc
pune
Nashik
Approve



All Students

Roll No	Name	Exam center
100	abc	Nashik
121	xyz	mumbai

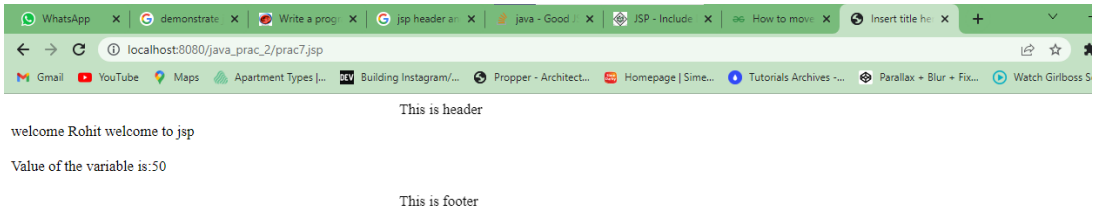
Practical No : 6f

Assignments based on web application development using JSP

6f.Write a JSP program that demonstrates the use of various JSP tags..

```
<%@ page import="java.util.Date"%>
<%@ 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>
        <%@ include file="header.jsp"%>
        <!-- scriplet tag -->
        <%!String name = "Rohit";%>
        <%
        out.print("welcome " + name);
        %>
        <!-- expression tag -->
        <%="welcome to jsp"%>
        <br>
        <br>
        <!-- declaration tag -->

        <%!int data = 50;%>
        <%="Value of the variable is:" + data%>
        <br>
        <br>
        <%@ include file="footer.jsp"%>
</body>
</html>
```

Output:-



PRACTICAL NO:7

Assignment based Spring Framework

Practical No : 7a

Assignment based Spring Framework

7a.WAP in Java using spring to demonstrate dependency injection via setter method .

Student.java File:-

```
package Animal;

public class Student {

   private String studentName;
   private String studentCourse;

   public String getStudentName()
   {
     return studentName;
   }

   public void setStudentName(String studentName)
   {
      this.studentName = studentName;
   }

   public String getStudentCourse()
   {
      return studentCourse;
   }

   public void setStudentCourse(String studentCourse)
   {
      this.studentCourse = studentCourse;
   }

   @Override public String toString()
   {
      return "Student{"
         + "studentName= Robert"  + studentName +
          ", studentCourse= MCA" + studentCourse + '}';
   }
}
```

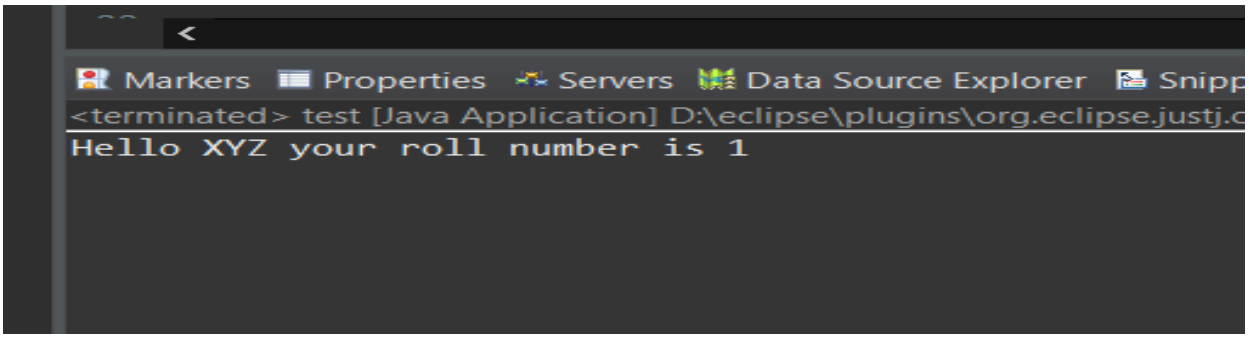
Main1.java File:-

```
package Animal;

import java.io.*;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class Main1 {
   public static void main(String[] args)
   {
         ApplicationContext context = new ClassPathXmlApplicationContext("config.xml");
      Student student= (Student)context.getBean("stud");
      System.out.println(student);
   }
}
```

Config.xml File

```
<?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:context="http://www.springframework.org/schema/context"
 xsi:schemaLocation="http://www.springframework.org/schema/beans
                    https://www.springframework.org/schema/beans/spring-beans.xsd
                   http://www.springframework.org/schema/context
                    http://www.springframework.org/schema/context/spring-context.xsd">

    <bean class="Animal.Student" name="stud">

   </bean>
</beans>
```

Output:-



Practical No : 7b

Assignment based Spring Framework

7b.WAP in Java using spring to demonstrate dependency injection via Constructor.

Main2.java File:-

```
package Animal;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class Main2 {
  public static void main(String[] args) {
    ApplicationContext context = new ClassPathXmlApplicationContext("Beans1.xml");

    TextEditor te = (TextEditor) context.getBean("textEditor");
    te.spellCheck();
  }
}
```

TextEditor.java File:-

```
package Animal;
public class TextEditor {
        private SpellChecker spellChecker;
        public TextEditor(SpellChecker spellChecker) {
          System.out.println("Inside TextEditor constructor." );
          this.spellChecker = spellChecker;
        }
        public void spellCheck() {
          spellChecker.checkSpelling();
        }
      }
```

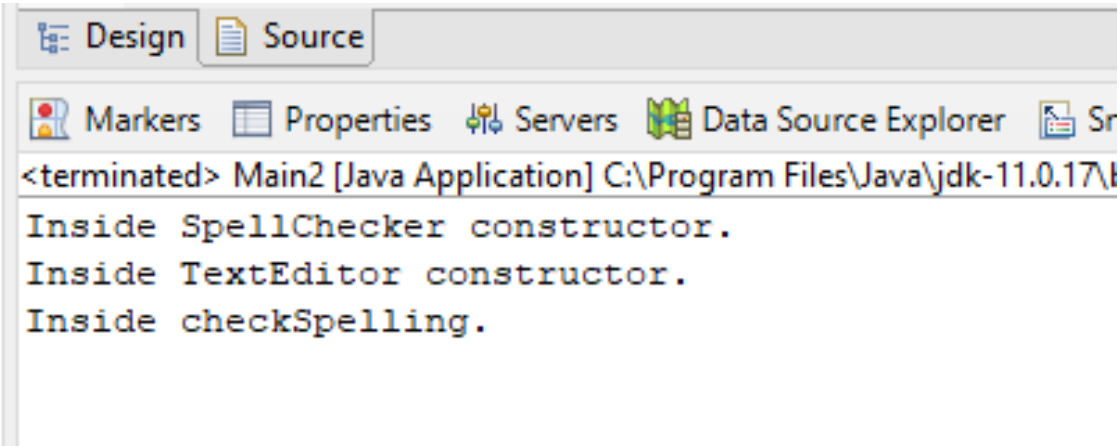
SpellChecker.java File:-

```
package Animal;
public class SpellChecker {
        public SpellChecker(){
          System.out.println("Inside SpellChecker constructor." );
        }
        public void checkSpelling() {
          System.out.println("Inside checkSpelling." );
        }}
```

Beans1.xml File:-

```
<?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-3.0.xsd">

  <!-- Definition for textEditor bean -->
  <bean id = "textEditor" class = "Animal.TextEditor">
    <constructor-arg ref = "spellChecker"/>
  </bean>

  <!-- Definition for spellChecker bean -->
  <bean id = "spellChecker" class = "Animal.SpellChecker"></bean>

</beans>
```

Output:-



Practical No : 7c

Assignment based Spring Framework

7c.WAP in Java using spring to demonstrate Autowiring.

B.java File:-

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

A1.java File:-

```
package Animal;
public class A1 {
B b;
A1(){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();
}
}
```

Cont.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.sssit.B"></bean>
<bean id="a" class="org.sssit.A" autowire="byName"></bean>

</beans>
```

Test2.java File :-

```
package Animal;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class Test2 {
public static void main(String[] args) {
   ApplicationContext context=new ClassPathXmlApplicationContext("cont.xml");
   A a=context.getBean("a",A.class);
   a.display();
}
}
```

Output:-

```
b is created
a is created
hello a
hello b
```

PRACTICAL NO:8

Assignment based Aspect Oriented Programming

Practical No. 8a

Assignment based Aspect Oriented Programming

8a.WAP to demonstrate Spring AOP – before advice.

A.java File:-

```
package Animal;
public class A
{
public void m()
{ System.out.println("actual business logic");
        System.out.println("Additional concern before actual logic"); }
}
```

BeforeAdvisor.java File:-

```
package Animal;
import java.lang.reflect.Method;
import org.springframework.aop.MethodBeforeAdvice;
public class BeforeAdvisor implements MethodBeforeAdvice
{ @Override public void before(Method method, Object[] args, Object target) throws Throwable
{
System.out.println("actual business logic");
System.out.println("Additional concern before actual logic");
}
}
```

applicationContext.xml File:-

```
<?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="obj" class="Animal.A"> </bean>
<bean id="ba" class="Animal.BeforeAdvisor"> </bean>
<bean id="proxy" class="org.springframework.aop.framework.ProxyFactoryBean">
<property name="target" ref="obj"> </property>
<property name="interceptorNames">
<list>

</list>
</property>
</bean>
</beans>
```

ProxyFactoryBean File:-

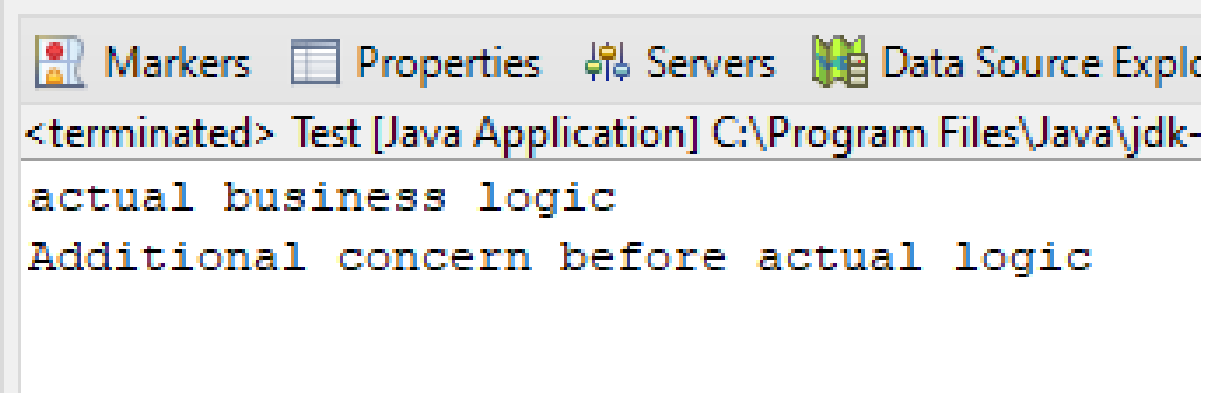
```
package Animal;
import java.util.List;
public class ProxyFactoryBean
{
 private Object target;
 private List interceptorNames;
//getters and setters
public Object getTarget() {
        return target;
}
public void setTarget(Object target) {
        this.target = target;
}
public List getInterceptorNames() {
        return interceptorNames;
}
public void setInterceptorNames(List interceptorNames) {
        this.interceptorNames = interceptorNames;
}
 }
```

Test.java File:-

```
package Animal;
 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");
BeanFactory factory=new XmlBeanFactory(r);
A a=factory.getBean("proxy",A.class);
a.m();
}
}
```

Output:-



Practical No. 8b

Assignment based Aspect Oriented Programming

8b.WAP to demonstrate Spring AOP – after advice.

Operation.java File:-

```
package Animal;
public class Operation
 {
 public void msg(){System.out.println("msg method invoked");}
 public int m(){System.out.println("m method invoked");return 2;}
 public int k(){System.out.println("k method invoked");return 3;}
}
```

applicationContext.xml File:-

```
<?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:aop="http://www.springframework.org/schema/aop"
 xsi:schemaLocation="http://www.springframework.org/schema/beans
 http://www.springframework.org/schema/beans/spring-beans.xsd
 http://www.springframework.org/schema/aop
 http://www.springframework.org/schema/aop/spring-aop.xsd">
<bean id="opBean" class="Animal.Operation"> </bean>
 <bean id="trackMyBean" class="Animal.TrackOperation"></bean>
<bean class="org.springframework.aop.aspectj.annotation.AnnotationAwareAspectJAutoP
proxyCreator"> </bean>
 </beans>
```

Test1.java File:-

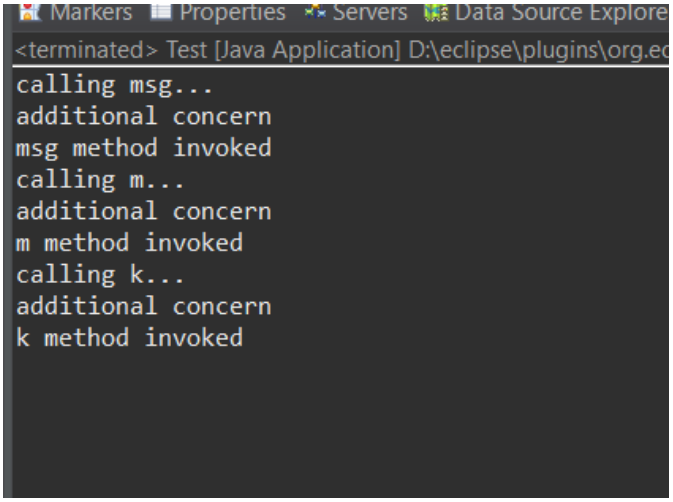
```
package Animal;
import org.springframework.context.ApplicationContext;
 import org.springframework.context.support.ClassPathXmlApplicationContext;
 public class Test1
 {
 public static void main(String[] args)
 {
ApplicationContext context = new
 ClassPathXmlApplicationContext("applicationContext.xml");
 Operation e = (Operation) context.getBean("opBean");
```

```
System.out.println("calling msg...");
e.msg();

System.out.println("calling m...");
e.m();

System.out.println("calling k...");
e.k();
}
}
```

TrackOperation.java File:-

```
package Animal;

//import org.aspectj.lang.JoinPoint;
 //import org.aspectj.lang.annotation.After;
 //import org.aspectj.lang.annotation.Pointcut;
 //@Aspect
 public class TrackOperation
{
 //@Pointcut("execution(* Operation.*(..))")
 public void k(){} //pointcut name
// @After("k()") //applying pointcut on after advice
 //public void myadvice(JoinPoint jp) //it is advice (after advice)
 {
 System.out.println("additional concern");
 //System.out.println("Method Signature: " + jp.getSignature());
 //}
```

Output:



Practical No. 8c

Assignment based Aspect Oriented Programming

8c.WAP to demonstrate Spring AOP – around advice.

A.javavFile:-

```
package Animal;
public class A
{
public void m()
{ System.out.println("actual business logic");
        System.out.println("Additional concern before actual logic"); }
}
```

AroundAdvisor.java File :-

```
package Animal;
 import org.aopalliance.intercept.MethodInterceptor;
 import org.aopalliance.intercept.MethodInvocation;
 public class AroundAdvisor implements MethodInterceptor
 {
 @Override
 public Object invoke(MethodInvocation mi) throws Throwable
{
 Object obj;
 System.out.println("additional concern before actual logic");
 obj=mi.proceed();
 System.out.println("additional concern after actual logic");
 return obj;
 }
 }
```

applicationContext.xml File :-

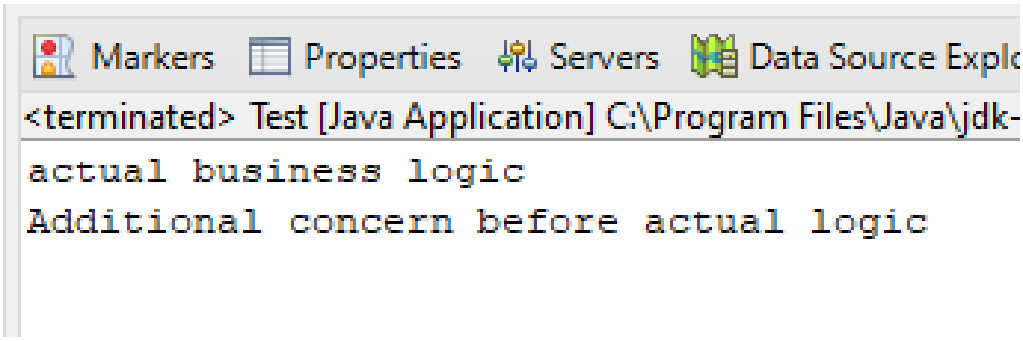
```
<?xml version="1.0" encoding="UTF-8"?>
 <beans
 xmlns="http://www.springframework.org/schema/beans"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:p="http://www.springframework.org/schema/p"
 xsi:schemaLocation="http://www.springframework.org/schema/beans
 http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">
```

```
<bean id="obj" class="Animal.A"> </bean>
<bean id="ba" class="Animal.AroundAdvisor"> </bean>
<bean id="proxy" class="org.springframework.aop.framework.ProxyFactoryBean">
<property name="target" ref="obj"> </property>
<property name="interceptorNames">
<list>
<value> ba </value>
</list>
</property>
</bean>
</beans>
```

Test1.java File:-

```
package Animal;
import org.springframework.context.ApplicationContext;
 import org.springframework.context.support.ClassPathXmlApplicationContext;
 public class Test1
 {
 public static void main(String[] args)
 {
ApplicationContext context = new
 ClassPathXmlApplicationContext("Appcontextt.xml");
 Operation e = (Operation) context.getBean("opBean");
 System.out.println("calling msg...");
 e.msg();
 System.out.println("calling m...");
 e.m();
 System.out.println("calling k...");
 e.k(); } }
```

Output:-



Practical No. 8d

Assignment based Aspect Oriented Programming

8d.WAP to demonstrate Spring AOP – after returning advice.

A.java File:-

```
package Animal;
public class A
{
public void m()
{ System.out.println("actual business logic");
        System.out.println("Additional concern before actual logic"); }
}
```

AfterAdvisor.java File:-

```
package Animal;
 import java.lang.reflect.Method;
 import org.springframework.aop.AfterReturningAdvice;
 public class AfterAdvisor implements AfterReturningAdvice
 {
@Override
 public void afterReturning(Object returnValue, Method method,
 Object[] args, Object target) throws Throwable
 { System.out.println("additional concern after returning advice"); }
 }
```

Context.xml File:-

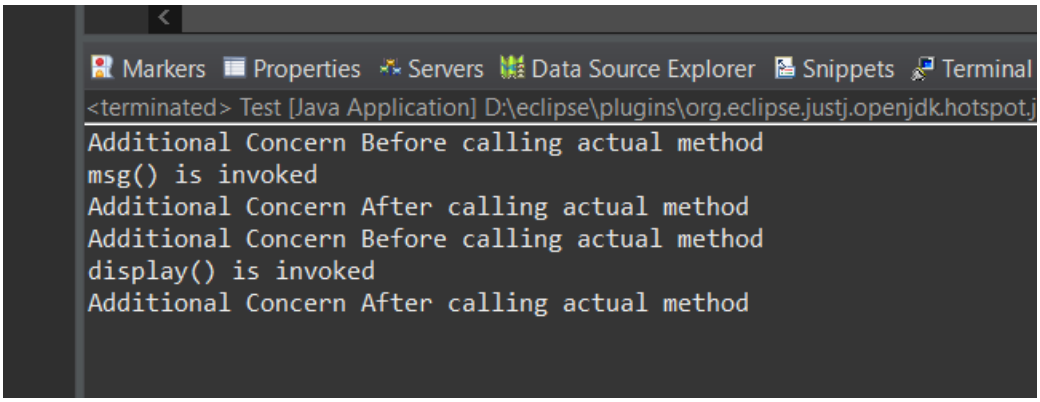
```
<?xml version="1.0" encoding="UTF-8"?>
 <beans
 xmlns="http://www.springframework.org/schema/beans"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:p="http://www.springframework.org/schema/p"
 xsi:schemaLocation="http://www.springframework.org/schema/beans>


 <bean id="obj" class="Animal.A"> </bean>
 <bean id="ba" class="Animalt.AfterAdvisor"> </bean>
 <bean id="proxy" class="org.springframework.aop.framework.ProxyFactoryBean">
 <property name="target" ref="obj"> </property>
 <property name="interceptorNames">
```

```
 <list>
 </list>
 </property>
 </bean>
</beans>
```

Test1.java File :-

```
package Animal
import org.springframework.context.ApplicationContext;
 import org.springframework.context.support.ClassPathXmlApplicationContext;
 public class Test1
 {
 public static void main(String[] args)
 {
ApplicationContext context = new ClassPathXmlApplicationContext("context.xml");
 Operation e = (Operation) context.getBean("opBean");
 System.out.println("calling msg...");
 e.msg();
 System.out.println("calling m...");
 e.m();
 System.out.println("calling k...");
 e.k(); }}
```

Output:-



PRACTICAL NO:9

Assignment based Spring JDBC

Practical No. 9a

Assignment based Spring JDBC

9a.Write a program to insert, update and delete records from the given table.

Employee.java File:-

```
package com.javatpoint;
public class Employee {
private int id;
private String name;
private float salary;
public Employee() {}
public Employee(int id, String name, float salary) {
        super();
        this.id = id;
        this.name = name;
        this.salary = salary;}
public int getId() {
        return id;}
public void setId(int id) {
        this.id = id;}
public String getName() {
        return name;}
public void setName(String name) {
        this.name = name;}
public float getSalary() {
        return salary;}
public void setSalary(float salary) {
        this.salary = salary;}}
```

EmployeeDao.java File:-

```
package com.javatpoint;

import org.springframework.jdbc.core.JdbcTemplate;

public class EmployeeDao {

private JdbcTemplate jdbcTemplate;

public void setJdbcTemplate(JdbcTemplate jdbcTemplate) {

        this.jdbcTemplate = jdbcTemplate;}

Public int saveEmployee(Employee e){

        String query="insert into employee
        values('"+e.getId()+"','"+e.getName()+"','"+e.getSalary()+"')";

        return jdbcTemplate.update(query);}

public int updateEmployee(Employee e){

        String query="update employee set name='"+e.getName()+"',salary='"+e.getSalary()+"'
        where id='"+e.getId()+"' ";

        return jdbcTemplate.update(query);}

public int deleteEmployee(Employee e){

        String query="delete from employee where id='"+e.getId()+"' ";

        return jdbcTemplate.update(query);}}
```

applicationContext.xml File:-

```
<?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="ds" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
```

```
<property name="driverClassName" value="com.mysql.jdbc.Driver" />
<property name="url" value="jdbc:mysql://localhost/employee" />
<property name="username" value="root" />
<property name="password" value="" />
</bean>
<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
<property name="dataSource" ref="ds"></property>
</bean>
<bean id="edao" class="com.javatpoint.EmployeeDao">
<property name="jdbcTemplate" ref="jdbcTemplate"></property>
</bean>
</beans>
```

Test.java File:-

```
package com.javatpoint;
import java.util.Scanner;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class Test {
public static void main(String[] args) {
        ApplicationContext ctx=new
ClassPathXmlApplicationContext("applicationContext.xml");
        EmployeeDao dao=(EmployeeDao)ctx.getBean("edao");
//      int status=dao.saveEmployee(new Employee(102,"Amit",35000));
        System.out.println("Type 1 to insert record");
        System.out.println("Type 2 to update record");
        System.out.println("Type 3 to delete record");
        Scanner sc =new Scanner(System.in);
```

```
int op=sc.nextInt();
while(op!=4) {
if(op==1) {
        System.out.println("enter id");
        int id=sc.nextInt();
        System.out.println("enter name");
        String name=sc.next();
        System.out.println("enter salary");
        int sal=sc.nextInt();
        int status=dao.saveEmployee(new Employee(id,name,sal));
        if (status>0) {
                System.out.println("Data inserted successfully");}
}else if(op==2) {
        System.out.println("enter id");
        int id=sc.nextInt();
        System.out.println("enter name");
        String name=sc.next();
        System.out.println("enter salary");
        int sal=sc.nextInt();
        int status=dao.updateEmployee(new Employee(id,name,sal));
        if (status>0) {
                System.out.println("Data updated successfully");}
}else if(op==3){
        System.out.println("enter id");
        int id=sc.nextInt();
        int status=dao.deleteEmployee(new Employee(id,"",0));
        System.out.println("Data deleted successfully");
}else {System.out.println("wrong input3");}
```

```
System.out.println("Type 1 to insert record");
System.out.println("Type 2 to update record");
System.out.println("Type 3 to delete record");
op=sc.nextInt();}        }}
```

Output :-

Insert data

```
Test (3) [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe
Type 1 to insert record
Type 2 to update record
Type 3 to delete record
1
enter id
100
enter name
abc
enter salary
35000
Data inserted successfully
```

Showing rows 0 - 0 (1 total, Query took 0.0009 seconds.)

```
SELECT * FROM `employee`
```

Show all | Number of rows: 25 Filter rows: Search this table

+ Options

id	name	salary
100	abc	35000

Update:-

```
Test (3) [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe
Type 1 to insert record
Type 2 to update record
Type 3 to delete record
2
enter id
100
enter name
abc
enter salary
100000
Data updated successfully
```

Showing rows 0 - 0 (1 total, Query took 0.0006 seconds.)

```
SELECT * FROM `employee`
```

Show all | Number of rows: 25 Filter rows: Search this table

+ Options

id	name	salary
100	abc	100000

Delete :-

```
Test (3) [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe
Type 1 to insert record
Type 2 to update record
Type 3 to delete record
3
enter id
100
Data deleted successfully
```

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0006 seconds.)

```
SELECT * FROM `employee`
```

id	name	salary

Query results operations

Create view

Practical No. 9b

Assignment based Spring JDBC

9b.Write a program to demonstrate PreparedStatement in Spring JdbcTemplate

Employee.java File:-

```
package com.javatpoint;
public class Employee {
private int id;
private String name;
private float salary;
public Employee() {}
public Employee(int id, String name, float salary) {
	super();
	this.id = id;
	this.name = name;
	this.salary = salary;}
public int getId() {
	return id;}
public void setId(int id) {
	this.id = id;}
public String getName() {
	return name;}
public void setName(String name) {
	this.name = name;}
public float getSalary() {
	return salary;}
public void setSalary(float salary) {
	this.salary = salary;}
}
```

applicationContext.xml File:-

```
<?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="ds" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
<property name="driverClassName" value="com.mysql.jdbc.Driver" />
<property name="url" value="jdbc:mysql://localhost/employee" />
<property name="username" value="root" />
<property name="password" value="" />
</bean>
<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
<property name="dataSource" ref="ds"></property>
</bean>
<bean id="edao" class="com.javatpoint.EmployeeDao">
<property name="jdbcTemplate" ref="jdbcTemplate"></property>
</bean>
</beans>
```

EmployeeDao.java File:-

```
package com.javatpoint;
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 EmployeeDao {
private JdbcTemplate jdbcTemplate;
public void setJdbcTemplate(JdbcTemplate jdbcTemplate) {
    this.jdbcTemplate = jdbcTemplate;  }
public Boolean saveEmployeeByPreparedStatement(final Employee e){
    String query="insert into employee values(?,?,?)";
    return jdbcTemplate.execute(query,new PreparedStatementCallback<Boolean>(){
    @Override
    public Boolean doInPreparedStatement(PreparedStatement ps)
            throws SQLException, DataAccessException {
        ps.setInt(1,e.getId());
        ps.setString(2,e.getName());
        ps.setFloat(3,e.getSalary());
        return ps.execute();  }  })}    }
```

Test.java File:-

```
package com.javatpoint;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class Test {
public static void main(String[] args) {
    ApplicationContext ctx=new
ClassPathXmlApplicationContext("applicationContext.xml");
    EmployeeDao dao=(EmployeeDao)ctx.getBean("edao");
    dao.saveEmployeeByPreparedStatement(new Employee(108,"Suresh",35000));
    System.out.println("Data inserted successfully using PreparedStatement");
}
}
```

Output:-

```
<terminated> Test (3) [Java Application] C:\Program Files\Java\jdk-14.0.2\bin\javaw.exe
Data inserted successfully using PreparedStatement
```

Showing rows 0 - 0 (1 total, Query took 0.0007 seconds.)

```
SELECT * FROM `employee`
```

Show all | Number of rows: 25 Filter rows: Search this table

+ Options

id	name	salary
108	Suresh	35000