# Practical N0: 01

## Assignment on Java Generics.

### Practical N0: 01A

1. **Write a java program to demonstrate a Generic class.**

package Generics;

class Generic<T>

{

T obj; Generic(T obj)

{

this.obj=obj;

}

public T getObject()

{

return this.obj;

}

}

public class Prac1 {

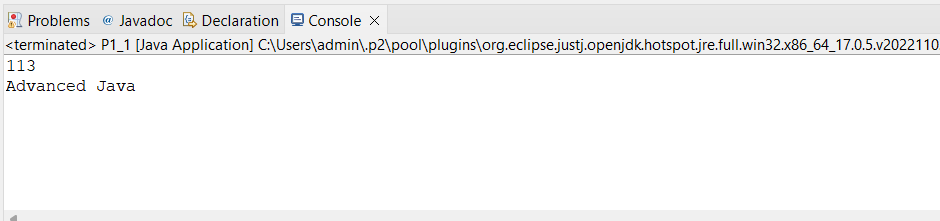
public static void main(String[] args) { Generic<Integer> g= new Generic(1234); System.out.println(g.getObject());

}

}

### Output:

Generic<String> G= new Generic("JAVA"); System.out.println(G.getObject());



### Practical N0: 01B

1. **Write a java program to demonstrate a Generic Methods.**

package Generics; class G\_M

{

static<T> void genericDisplay(T Element)

{

System.*out*.println(Element.getClass().getName()+"="+ Element);

}

}

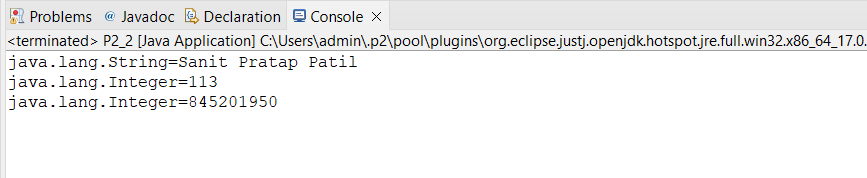
public class Prac2 {

public static void main(String[] args) { G\_M.*genericDisplay*(100); G\_M.*genericDisplay*("Siddhesh"); G\_M.*genericDisplay*(100000l);

}

}

### Output:



**Practical N0: 01C**

### Write a java program to demonstrate Wildcard in java Generic.

package Generics; import java.util.\*; public class WildCard {

private static double sum(List<?extends Number>list)

{

double sum = 0; for(Number i:list)

{

sum=sum+i.doubleValue();

}

return sum;

}

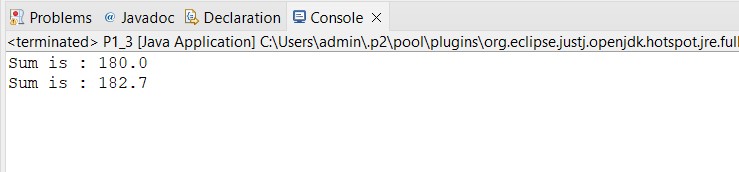
public static void main(String[] args) { List<Integer> a=Arrays.*asList*(10,20,30,40); System.*out*.println("Sum Is : "+*sum*(a));

List<Double> b=Arrays.*asList*(10.1,20.1,30.1,40.2); System.*out*.println("Sum Is : "+*sum*(b));

}

}

**Output:**



# Practical N0: 02

## Assignment on List Interface.

### Practical N0: 02A

1. **Write a java program to create List containing list of items of type String and use for—each loop to print item of the list.**

package List; import java.util.\*;

public class Pract1 {

public static void main(String[] args) {

List <String> l1= new ArrayList<>();

l1.add("Siddhesh");

l1.add("100");

l1.add("FY MCA");

// for each loop for(String i:l1)

{

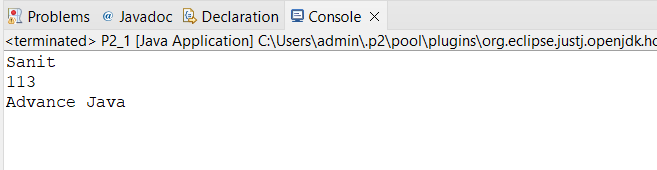
System.*out*.println(i);

}

}

}

### Output:



**Practical N0: 02B**

### Write a java program to create List containing list of items and use List Iterator interface To print items present in the list . Also print the list in revers direction.

package List; import java.util.\*;

public class ListIteratorr {

public static void main(String[] args) { List <String> l1= new ArrayList<>();

l1.add("Siddhesh");

l1.add("100");

l1.add("FY MCA");

ListIterator<String> l1Iterator = l1.listIterator(); System.*out*.println("-------using Iterator ");

while (l1Iterator.hasNext()) { System.*out*.println(l1Iterator.next());

}

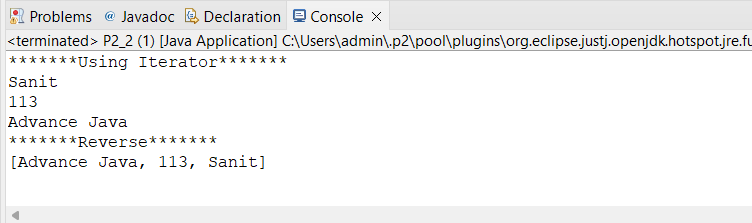
System.*out*.println("------Reverse ");

Collections.*reverse*(l1); System.*out*.println(l1);

}

}

**Output:**



# Practical N0: 03

## Assignment on Set Interface.

### Practical N0: 03A

1. **Write a java program to create Set containing list of items of type String and print the items in the list using iterator interface. Also print the list in reverse direction.**

package Pract3; import java.util.\*;

public class SetiI\_f {

public static void main(String[] args) {

Set<String> s = new TreeSet<String>(); s.add("Pratikesh");

s.add("Siddesh");

s.add("Sanith");

s.add("Sanket"); System.*out*.println(s);

Iterator<String> l1Iterator = s.iterator(); System.*out*.println("-------using Iterator ");

while (l1Iterator.hasNext())

{

System.*out*.println(l1Iterator.next());

}

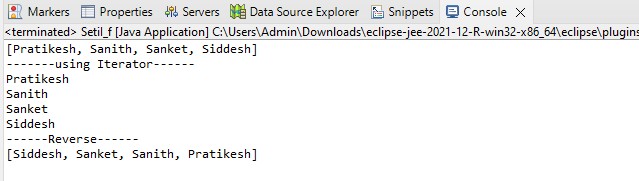
System.*out*.println("------Reverse ");

NavigableSet<String>treereverse = ((TreeSet<String>) s).descendingSet(); System.*out*.println(treereverse);

}

}

### Output:



**Practical N0: 03B**

### Write a java program using set interface containing list of items and perform the following operations.

1. **Add items in the set.**
2. **Insert items of one set in to other set.**
3. **Remove items from the set.**
4. **Search the specific item in the set.**

package Pract3; import java.util.\*;

public class Demo {

public static void main(String[] args) {

Set name = new HashSet<>(); name.add("Siddhesh"); //a name.add("Pratikesh"); name.add("Sanit");

name.add("Samket"); System.*out*.println(name);

Set num = new HashSet<>(); num.add(100);

num.add(73);

num.add(113);

num.add(94); System.*out*.println(num);

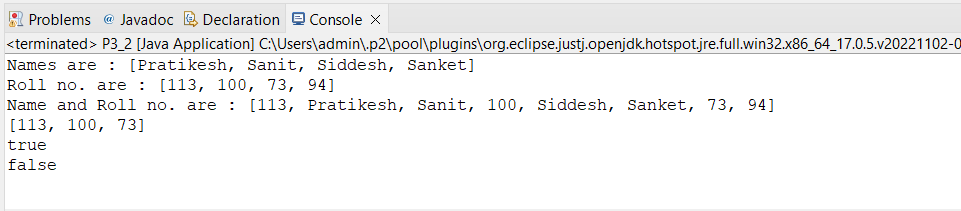
name.addAll(num); //b System.*out*.println(name);

num.remove(94); //c System.*out*.println(num); System.*out*.println(name.contains("Siddhesh"));//d

}

}

**Output:**



# Practical N0: 04

## Assignment on Map Interface.

### Practical N0: 04A

A. **Write a java program using Map containing list of items having keys and associated values and perform the following operations.**

* 1. **Add items in the map.**
  2. **Remove items from the map.**
  3. **Search specific key from the map.**
  4. **Get value of specific key.**
  5. **Insert map element of one map in to another map.**
  6. **Print all keys and values of the map.**

package Pract\_4; import java.util.\*;

public class Map\_if {

public static void main(String[] args) {

Map<String, Integer> marks = new HashMap<>(); marks.put("java", 95);

marks.put("c", 93);

marks.put("python", 95);

marks.put("adv java", 98); System.*out*.println(marks);

Map<String, Integer> stud = new HashMap<>(); stud.put("Siddhesh", 100);

stud.put("Practikesh", 73);

stud.put("Sanit", 113);

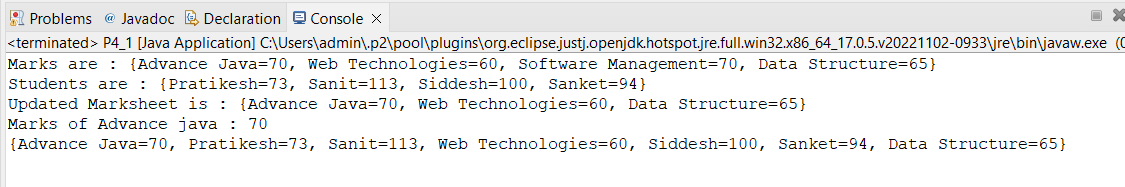
stud.put("Snket", 93); System.*out*.println(stud);

marks.remove("c"); System.*out*.println(marks); System.*out*.println(marks.get("java")); marks.putAll(stud); System.*out*.println(marks);

}

}

**Output:**



# Practical N0: 05

## Assignment on Lambda Expression.

### Practical N0: 05A

1. **Write a java program using Lambda Expression with single parameter.**

package Pract\_5; interface FunInterface1

{

void abstractfun(int a);

}

public class SingleParameter {

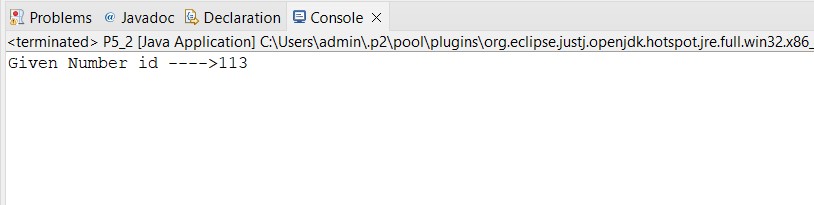
public static void main(String[] args) {

FunInterface1 fobj=(int a)->System.*out*.println("Given Number id --> "+a); fobj.abstractfun(100);

}

}

### Output:



**Practical N0: 05B**

### Write a java program using Lambda Expression with multiple parameters to add two numbers.

package Pract\_5; interface FunInterface2

{

void abstractfun(int x,int y);

}

public class Multi\_Parameter {

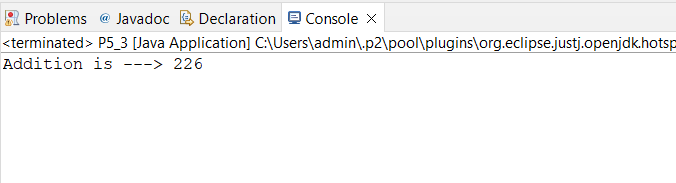
public static void main(String[] args) {

FunInterface2 fobj=(int x,int y)->System.*out*.println("Addition is --> "+(x+y)); fobj.abstractfun(50,50);

}

}

### Output:



**Practical N0: 05C**

### Write a Java program using Lambda Expression to Convert Fahrenheit to Celsius.

package Pract\_5; interface Conversion

{

void xyz(float a);

}

public class Convert {

public static void main(String[] args) {

Conversion obj= (float a)->

{

float fahr=(a\*1.8f)+30; System.*out*.println(fahr);

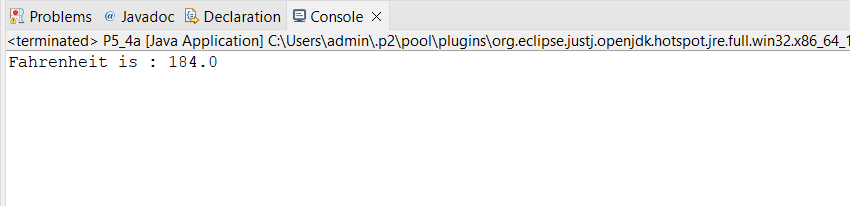
};

obj.xyz(20f);

}

}

### Output:



**Practical N0: 05D**

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

package Pract\_5;

// Program for Check the number is even or odd interface EvenOdd

{

void abc(int x);

}

public class Program {

public static void main(String[] args) { EvenOdd EO=(int x)->

{

if(x>=0)

{

}

else

{

}

};

System.*out*.println(x+" is Even Number");

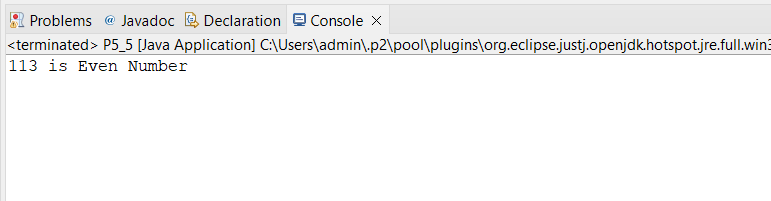
System.*out*.println(x+" is odd Number");

EO.abc(100);

}

}

### Output:



**Practical N0: 05E**

### Write a Java program using Lambda Expression to concatenate two String.

package Pract\_5; interface concat

{

String abc(String s1,String s2);

}

public class Concatenation {

public static void main(String[] args) { concat C =(String s1,String s2)->

{

return(s1+s2);

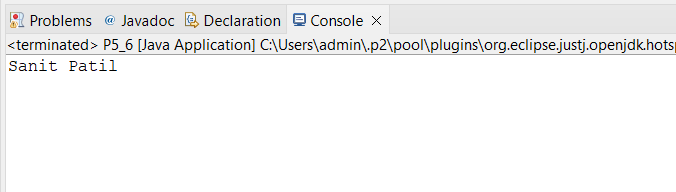
};

System.*out*.println(C.abc("Siddhesh", " More"));

}

}

**Output:**



# Practical N0: 06

## Assignment based on web application development using JSP.

### Practical N0: 06A

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

<%@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>

Name">

<tr>

</tr>

<tr>

</tr>

<td>Enter Name:</td>

<td><input type="text" name="T2" placeholder="Enter Your

</td>

<td>Enter Number:</td>

<td><input type="tel" name="T3" placeholder="Enter Mobile No">

</td>

<input type="text" name="task" value="insert" hidden>

<tr>

</tr>

<td>

</td>

<td>

</td>

<button type="reset">Reset</button>

<button type="submit">Insert</button>

</table>

</form>

<form name="B" method="post" action="index.jsp">

<input type="text" name="task" value="show" hidden>

<table>

<tr>

<button type="submit">Show</button>

</td>

</tr>

</table>

</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 = "root";

String url = "jdbc:mysql://localhost:3307/tel\_db";

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 = "root";

String url = "jdbc:mysql://localhost:3307/tel\_db";

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();

}

}

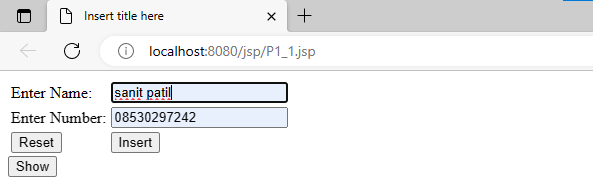
}

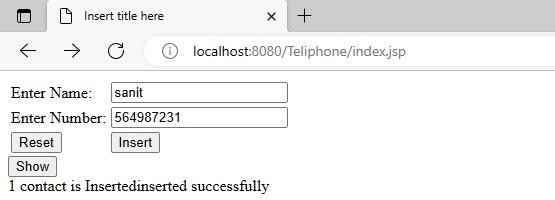
%>

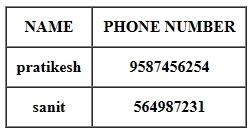
</body>

</html>

### Output:







**Practical N0: 06B**

### Write a jsp page 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>

</tr>

<tr>

</tr>

<tr>

</tr>

</tr>

<tr>

</tr>

<tr>

<td>First Name</td>

<td><input type=*"text"* name=*"first\_name"* /></td>

<td>Last Name</td>

<td><input type=*"text"* name=*"last\_name"* /></td>

<td>UserName</td>

<td><input type=*"text"* name=*"username"* /></td>

<tr>

<td>Password</td>

<td><input type=*"password"* name=*"password"* /></td>

<td>Address</td>

<td><input type=*"text"* name=*"address"* /></td>

<td>Contact No</td>

<td><input type=*"text"* name=*"contact"* /></td>

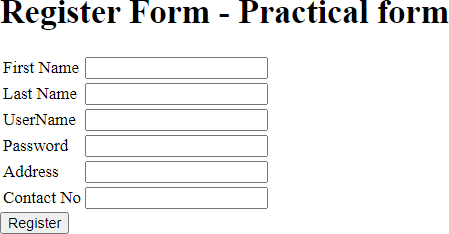
</body>

</html>

</tr></table>

<input type=*"submit"* value=*"Register"* /></form>

### Output:



**Practical N0: 06C**

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

1. **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"*>

*Nmae"*>

<table>

<tr>

</tr>

<tr>

</tr>

<tr>

</tr>

<tr>

</tr>

<td>Roll No:</td>

<td><input type=*"number"* name=*"T1"* placeholder=*"Enter Your Roll No"*></td>

<td>Name:</td>

<td><input type=*"text"* name=*"T2"* placeholder=*"Enter Your*

</td>

<td>Semester:</td>

<td><input type=*"text"* name=*"T3"* placeholder=*"Enter Semester"*>

</td>

<td>Course:</td>

<td><input type=*"text"* name=*"T4"* placeholder=*"Enter Your Course"*></td>

<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 = "root";

String url = "jdbc:mysql://localhost:3307/SM\_db";

Connection con = DriverManager.getConnection(url, username, password);

//Connection con= DriverManager.getConnection("SM\_db","root","root"); try {

String q = "insert into student 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 = "root";

String url = "jdbc:mysql://localhost:3307/SM\_db";

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 student 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 = "root";

String url = "jdbc:mysql://localhost:3307/SM\_db";

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 = "root";

String url = "jdbc:mysql://localhost:3307/SM\_db";

Connection con = DriverManager.getConnection(url, username, password); ResultSet rs;

String query = "select \* from student"; 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 = "root";

String url = "jdbc:mysql://localhost:3307/SM\_db";

Connection con = DriverManager.getConnection(url, username, password); ResultSet rs;

String query = "select \* from student where id=" + 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>

</tr>

<tr>

</tr>

<tr>

<td>Name:</td>

<td><input type=*"text"* name=*"T2"* value=*"*<%=rs.getString(2)%>*"*>

</td>

<td>Semester:</td>

<td><input type=*"text"* name=*"T3"* value=*"*<%=rs.getString(3)%>*"*>

</td>

<td>Course:</td>

<td><input type=*"text"* name=*"T4"* value=*"*<%=rs.getString(4)%>*"*>

</td>

</tr>

<input type=*"text"* name=*"task"* value=*"update"* hidden>

<tr>

</tr>

</table>

</form>

<%

}

} **catch** (Exception e) { e.printStackTrace();

}

}

%>

</body>

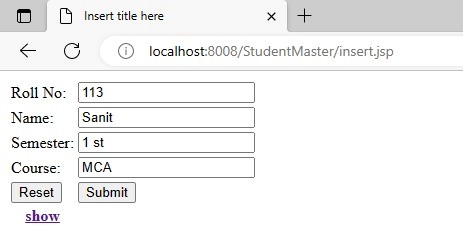
</html>

### Output:

* 1. **Insert**

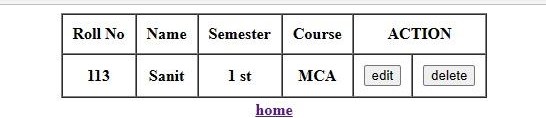
<td><input type=*"reset"*></td>

<td><input type=*"submit"*></td>

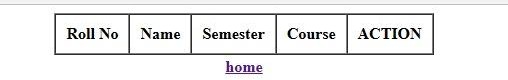


### Show

* 1. **Update**



### Delete



**Practical N0: 06D**

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

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

### 8 to 15 year at 5.5%

1. **16 to 30 year at 5.75%**

**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="index.jsp" method="post">

<table style="with: 50%">

<tr>

</tr>

<tr>

</tr>

</table>

<td>Enter Principal amount</td>

<td><input type="number" name="pa" /></td>

<td>Enter time period (In years)</td>

<td><input type="number" name="ti" /></td>

<td><input type="text" name="task" value="cal" hidden /></td>

<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);

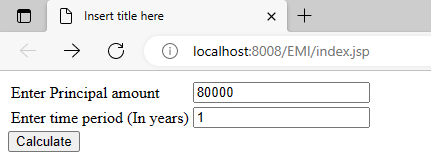
}

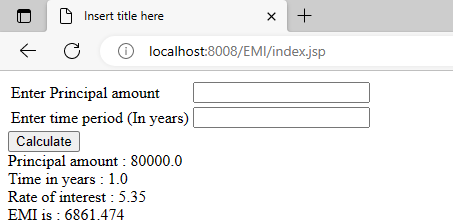
%>

</body>

</html>

### Output:





**Practical N0: 06E**

### Write a program using jsp that displays a webpage consisting application form for change of study centre which can be filled by any student who wants to change his/her study centre.

<%@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=*"prac5.jsp"*>

<table>

<tr>

</tr>

<tr>

</tr>

<tr>

</tr>

<tr>

</tr>

<td>Roll No:</td>

<td><input type=*"number"* name=*"T1"* placeholder=*""*></td>

<td>Name:</td>

<td><input type=*"text"* name=*"T2"* placeholder=*""*></td>

<td>Previous exam center:</td>

<td><input type=*"text"* name=*"T3"* placeholder=*""*></td>

<td>New exam center:</td>

<td><input type=*"text"* name=*"T4"* placeholder=*""*></td>

<input type=*"text"* name=*"task"* value=*"insert"* hidden>

<tr>

</tr>

<td><input type=*"submit"* value=*"Send Request"*></td>

<br>

</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/prac5", "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");

#### 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>

<input

<th>

<form action=*"prac5.jsp"* method=*"post"*>

<input name=*"task"* value=*"approve\_req"* hidden>

name=*"roll\_no"*

value=*"*<%=rs.getString(2)%>*"* hidden> <input hidden> <input value=*"*<%=rs.getString(1)%>*"* hidden> <input

name=*"nc"* value=*"*<%=rs.getString(5)%>*"* name=*"request\_id"*

type=*submit* value=*"Approve"* />

</tr>

<%

}

%>

</tbody>

</table>

<%

} **catch** (Exception e) { e.printStackTrace();

</th>

</form>

}

}

**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");

}

}

%>

<%

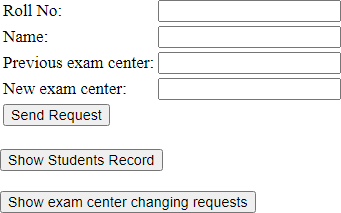
}

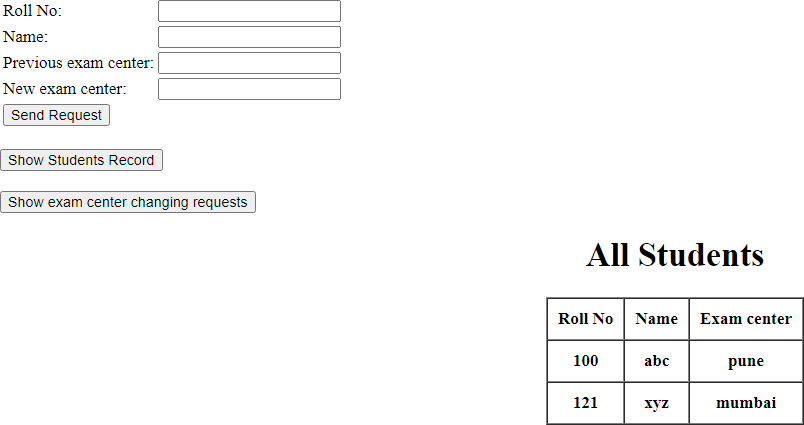
%>

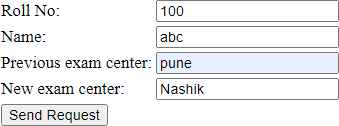
</body>

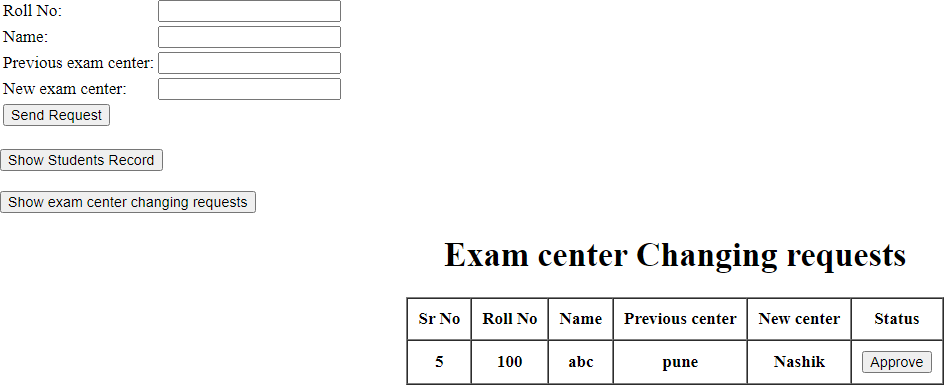
</html>

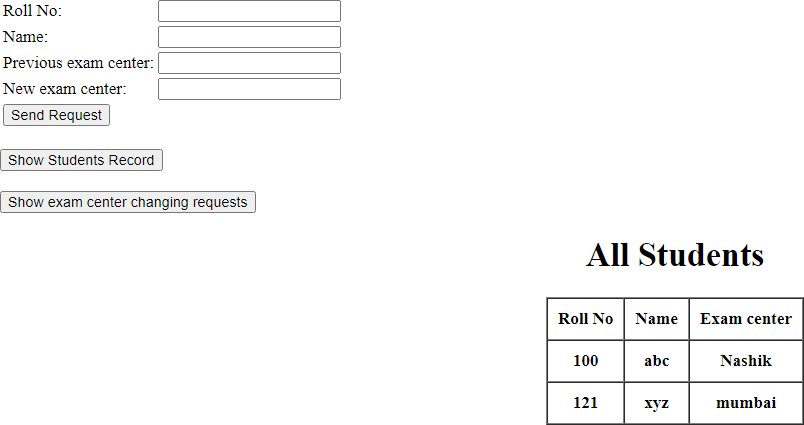
**Output:**











### Practical N0: 06F

1. **Write a JSP program that demonstrate the use of jsp declaration, Scriptlet directives, expression header and footer.**

<%@ 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 = "Siddhesh";%>

<%

out.print("I am " + name);

%>

<!-- expression tag -->

<%="welcome to jsp"%>

<br>

<br>

<!-- declaration tag -->

<%!**int** data = 100;%>

<%="Value of the variable is:" + data%>

<br>

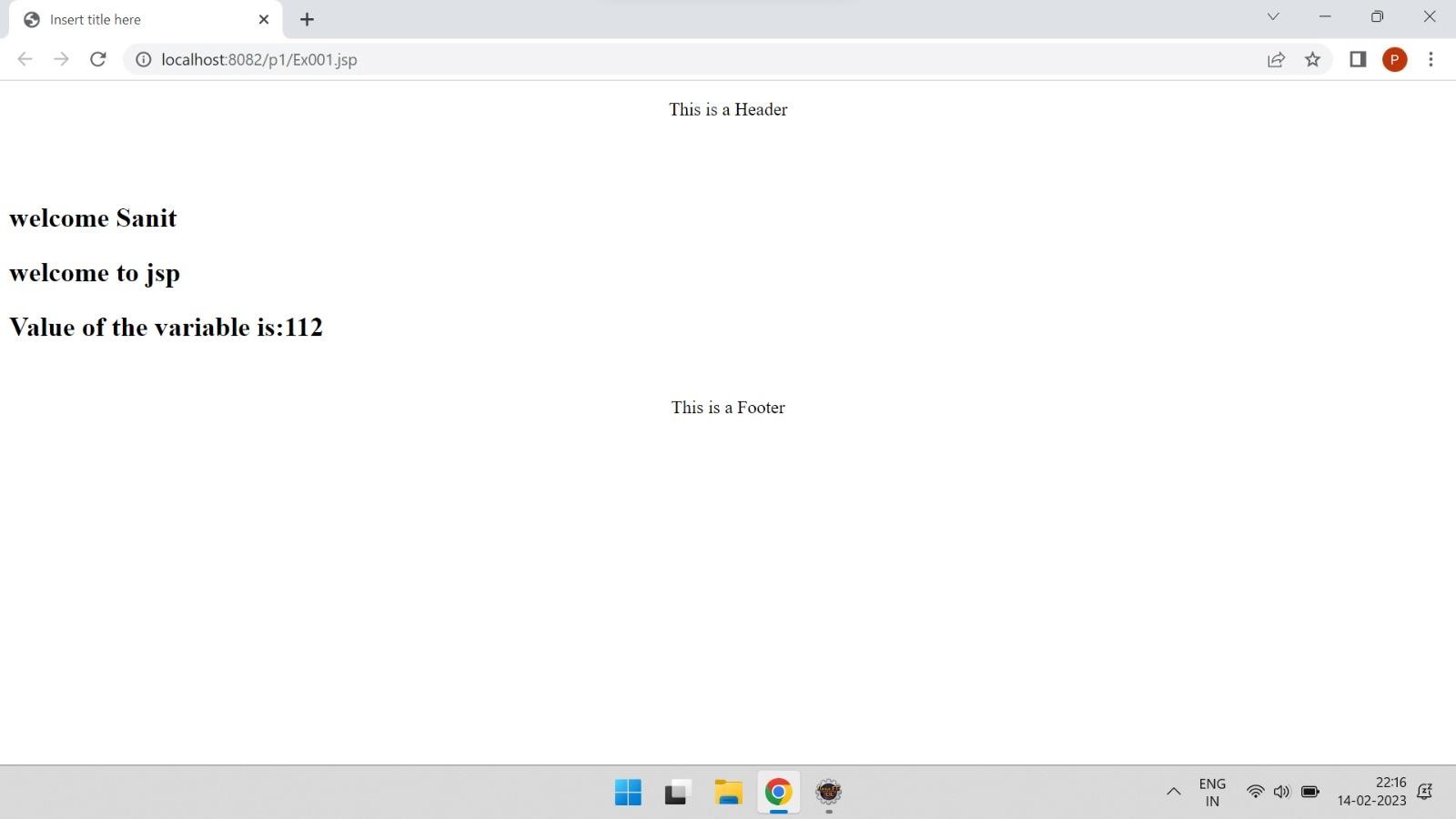
<br>

<%@ include file=*"footer.jsp"*%>

</body>

</html>

**Output:**



# Practical N0: 07

## Assignment based Spring Framework.

### Practical N0: 07A

1. **Write a java using spring to demonstrate dependency injection via setter method.**

### Student.java

package com;

public class Student { int x;

String y;

public int getX() {

return x;

}

public void setX(int x) { this.x = x;

}

public String getY() {

return y;

}

public void setY(String y) { this.y = y;

}

public void show()

{

System.*out*.println("Hello"); System.*out*.println("Roll No: " + x+" Name "+y);

}

}

### Test.java

package com;

import org.springframework.context.ApplicationContext;

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

@SuppressWarnings("resource") public static void main(String[] args) {

// TODO Auto-generated method stub

ApplicationContext A = new ClassPathXmlApplicationContext("demo.xml"); Student S =(Student)A.getBean("S1");

S.show();

}

}

### Demo.xml

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

<beans xmlns=*"*[*http://www.springframework.org/schema/beans"*](http://www.springframework.org/schema/beans)xmlns:xsi=*"*[*http://www.w3.org/2001/XMLSchema-instance*](http://www.w3.org/2001/XMLSchema-instance)*"* xmlns:aop=*"*[*http://www.springframework.org/schema/aop"*](http://www.springframework.org/schema/aop)xsi:schemaLocation=*"*[*http://www.springframework.org/schema/beans*](http://www.springframework.org/schema/beans)[*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*](http://www.springframework.org/schema/beans/spring-beans-3.0.xsd)*"*>

<bean id=*"S1"* class=*"com.Student"*>

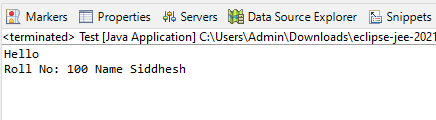
<property name=*"100"* ref=*"x"*></property>

<property name=*"Siddhesh"* ref=*"y"*></property>

</bean>

</beans>

### Output:



**Practical N0: 07B**

### Write a java using spring to demonstrate dependency injection via Constructor.

**Student.java**

package com;

public class Student { int x;

String y;

public Student(int x, String y) { super();

this.x = x; this.y = y;

}

public void setY(String y) { this.y = y;

}

public void show()

{

System.*out*.println("Hello"); System.*out*.println("Roll No: " + x+" Name "+y);

}

}

### Test.java

package com;

import org.springframework.context.ApplicationContext;

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

@SuppressWarnings("resource") public static void main(String[] args) {

// TODO Auto-generated method stub

ApplicationContext A = new ClassPathXmlApplicationContext("demo.xml");

Student S =(Student)A.getBean("S1"); S.show();

//Student student=(Student)A.getBean("S1");

//student.show();

}

}

### Demo.xml

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

<beans xmlns=*"*[*http://www.springframework.org/schema/beans"*](http://www.springframework.org/schema/beans)xmlns:xsi=*"*[*http://www.w3.org/2001/XMLSchema-instance*](http://www.w3.org/2001/XMLSchema-instance)*"* xmlns:aop=*"*[*http://www.springframework.org/schema/aop"*](http://www.springframework.org/schema/aop)xsi:schemaLocation=*"*[*http://www.springframework.org/schema/beans*](http://www.springframework.org/schema/beans)[*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*](http://www.springframework.org/schema/beans/spring-beans-3.0.xsd)*"*>

<bean id=*"S1"* class=*"com.Student"*>

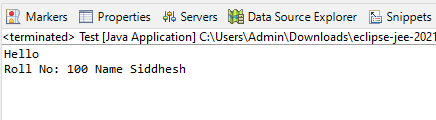
<constructor-arg value=*"100"* type=*"int"*></constructor-arg>

<constructor-arg value=*"Siddhesh"* type=*"String"*></constructor-arg>

</bean>

</beans>

### Output:



**Practical N0: 07C**

### Write a java using spring to demonstrate dependency Autowiring.

**Address.java**

package org.learn; public class Address {

private String RoadNm, city;

public String getRoadNm() { return RoadNm;}

public void setRoadNm(String roadNm) { RoadNm = roadNm;}

public String getCity(){ return city;}

public void setCity(String city) { this.city = city; }

public Address(String roadNm, String city)

{

super();

RoadNm = roadNm; this.city = city;

}

public Address() { super();} @Override

public String toString()

{ return "Address [RoadNm=" + RoadNm + ", city=" + city + "]"; }

void show()

{

System.*out*.println(RoadNm +"");

}

}

### Employee.java

package org.learn; public class Employee {

private int rollno; private Address addr;

public int getRollno() { return rollno;}

public void setRollno(int rollno) { this.rollno = rollno;}

public Address getAddr() { return addr;}

public void setAddr(Address addr) { this.addr = addr;}

public Employee(int rollno, Address addr)

{

super();

this.rollno = rollno;

this.addr = addr;

}

public Employee() { super();}

@Override

public String toString()

{ return "Employee [rollno=" + rollno + ", addr=" + addr + "]"; } void show()

{

System.*out*.println(rollno +" "+addr);

}

}

### Test.java

package org.learn;

import org.springframework.context.ApplicationContext;

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

public static void main(String[] args) {

ApplicationContext app = new ClassPathXmlApplicationContext("Autowire2.xml"); Employee E =(Employee)app.getBean("D");

E.show();

}

}

### Autowire2.xml

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

<beans xmlns=*"*[*http://www.springframework.org/schema/beans"*](http://www.springframework.org/schema/beans)xmlns:xsi=*"*[*http://www.w3.org/2001/XMLSchema-instance*](http://www.w3.org/2001/XMLSchema-instance)*"* xmlns:p=*"*[*http://www.springframework.org/schema/p"*](http://www.springframework.org/schema/p)xsi:schemaLocation=*"*[*http://www.springframework.org/schema/beans*](http://www.springframework.org/schema/beans)[*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*](http://www.springframework.org/schema/beans/spring-beans-3.0.xsd)*"*>

<bean id=*"A"* class=*"org.learn.Address"*>

<constructor-arg value=*"Raj Path"* type=*"String"*></constructor-arg>

<constructor-arg value=*"Delhi"* type=*"String"*></constructor-arg>

</bean>

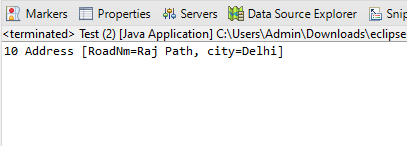
<bean id=*"D"* class=*"org.learn.Employee"* autowire=*"constructor"*>

<constructor-arg value=*"10"* type=*"int"*></constructor-arg>

</bean>

</beans>

**Output:**



# Practical N0: 08

## Assignment based Aspect Oriented Programming.

### Practical N0: 08A

1. **Write a program to demonstrate Spring AOP – before advice**

### Test.java:

package com.mba;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Test {

public static void main(String[] args) { ApplicationContext context = new

ClassPathXmlApplicationContext("applicationContextmba.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

package com.mba;

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 TrackOperation { @Pointcut("execution(\* Operation.\*(..))") public void k() {

}// pointcut name

@Before("k()") // applying pointcut on before advice

public void myadvice(JoinPoint jp)// it is advice (before advice)

{

System.*out*.println("additional concern");

//System.out.println("Method Signature: " + jp.getSignature());

}

}

### Operations.java:

package com.mba; 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;

}

}

### applicationContextmba.xml:

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

<beans xmlns=[*"http://*](http://www.springframework.org/schema/beans)*w*[*ww.springframework.org/schema/beans*](http://www.springframework.org/schema/beans)*"* xmlns:xsi=[*"http://*](http://www.w3.org/2001/XMLSchema-instance)*w*[*ww.w3.org/2001/XMLSchema-instance*](http://www.w3.org/2001/XMLSchema-instance)*"* xmlns:aop=[*"http://*](http://www.springframework.org/schema/aop)*w*[*ww.springframework.org/schema/aop"*](http://www.springframework.org/schema/aop)xsi:schemaLocation=[*"http://*](http://www.springframework.org/schema/beans)*w*[*ww.springframework.org/schema/beans*](http://www.springframework.org/schema/beans)[*http://www.springframework.org/schema/beans/spring-beans.xsd*](http://www.springframework.org/schema/beans/spring-beans.xsd)[*http://www.springframework.org/schema/aop*](http://www.springframework.org/schema/aop) [*http://www.springframework.org/schema/aop/spring-*](http://www.springframework.org/schema/aop/spring-aop.xsd)[*aop.xsd"*](http://www.springframework.org/schema/aop/spring-aop.xsd)>

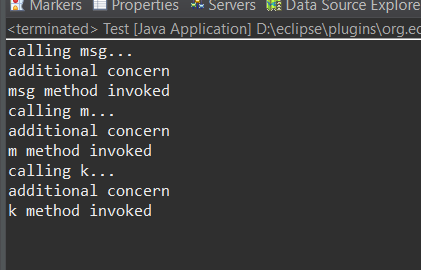
<bean id=*"opBean"* class=*"com.mba.Operation"*> </bean>

<bean id=*"trackMyBean"* class=*"com.mba.TrackOperation"*></bean>

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

</beans>

### Output:



**Practical N0: 08b**

### Write a program to demonstrate Spring AOP – after advice.

**Test.java:**

package com.mba;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Test {

public static void main(String[] args) { ApplicationContext context = new

ClassPathXmlApplicationContext("applicationContextmba.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

package com.mba;

import org.aspectj.lang.JoinPoint;

import org.aspectj.lang.annotation.Aspect; 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());

}

}

### Operations.java:

package com.mba; 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;

}

}

### applicationContextmba.xml:

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

<beans xmlns=[*"http://*](http://www.springframework.org/schema/beans)*w*[*ww.springframework.org/schema/beans*](http://www.springframework.org/schema/beans)*"* xmlns:xsi=[*"http://*](http://www.w3.org/2001/XMLSchema-instance)*w*[*ww.w3.org/2001/XMLSchema-instance*](http://www.w3.org/2001/XMLSchema-instance)*"* xmlns:aop=[*"http://*](http://www.springframework.org/schema/aop)*w*[*ww.springframework.org/schema/aop"*](http://www.springframework.org/schema/aop)xsi:schemaLocation=[*"http://*](http://www.springframework.org/schema/beans)*w*[*ww.springframework.org/schema/beans*](http://www.springframework.org/schema/beans)[*http://www.springframework.org/schema/beans/spring-beans.xsd*](http://www.springframework.org/schema/beans/spring-beans.xsd)[*http://www.springframework.org/schema/aop*](http://www.springframework.org/schema/aop) [*http://www.springframework.org/schema/aop/spring-*](http://www.springframework.org/schema/aop/spring-aop.xsd)[*aop.xsd"*](http://www.springframework.org/schema/aop/spring-aop.xsd)>

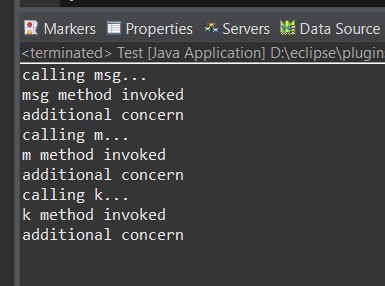
<bean id=*"opBean"* class=*"com.mba.Operation"*> </bean>

<bean id=*"trackMyBean"* class=*"com.mba.TrackOperation"*></bean>

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

</beans>

### Output:



**Practical N0: 08C**

### Write a program to demonstrate Spring AOP – around advice.

**Operation2.java:**

package com.mba;

public class Operation2 {

public void msg() {

System.*out*.println("msg() is invoked");

}

public void display() {

System.*out*.println("display() is invoked");

}

}

### TrackOperation2.java:

package com.mba;

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 TrackOperation2 { @Pointcut("execution(\* Operation2.\*(..))") public void abcPointcut() {

}

@Around("abcPointcut()")

public Object myadvice(ProceedingJoinPoint pjp) throws Throwable { System.*out*.println("Additional Concern Before calling actual method"); Object obj = pjp.proceed();

System.*out*.println("Additional Concern After calling actual method"); return obj;

}

}

### Test.java

package com.mba;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Test {

public static void main(String[] args) { ApplicationContext context = new

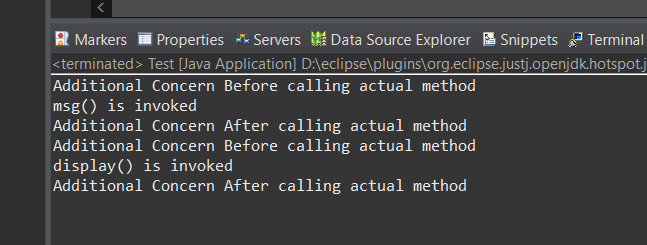
ClassPathXmlApplicationContext("applicationContextmba.xml"); Operation2 op = (Operation2) context.getBean("opBean"); op.msg();

op.display();

}

}

### Output:



**Practical N0: 08D**

### Write a program to demonstrate Spring AOP – after returning advice.

**Operation.java:**

package com.mbr;

public class Operation {

public int m() {

System.*out*.println("m() method invoked"); return 2;

}

public int k() {

System.*out*.println("k() method invoked"); return 3;

}

}

### TrackOperation:

package com.mbr;

import org.aspectj.lang.JoinPoint;

import org.aspectj.lang.annotation.AfterReturning; import org.aspectj.lang.annotation.Aspect;

@Aspect

public class TrackOperation {

@AfterReturning(pointcut = "execution(\* Operation.\*(..))", returning = "result") public void myadvice(JoinPoint jp, Object result)// it is advice (after returning advice)

{

System.*out*.println("additional concern"); System.*out*.println("Method Signature: " + jp.getSignature()); System.*out*.println("Result in advice: " + result); System.*out*.println("end of after returning advice...");

}

}

### Test.java:

package com.mbr;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Test {

public static void main(String[] args) { ApplicationContext context = new

ClassPathXmlApplicationContext("applicationContextmbr.xml"); Operation e = (Operation) context.getBean("opBean"); System.*out*.println("calling m..."); System.*out*.println(e.m());

System.*out*.println("calling k..."); System.*out*.println(e.k());

}

}

### applicationContextmbr.xml:

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

<**beans** xmlns=[*"http://*](http://www.springframework.org/schema/beans)*w*[*ww.springframework.org/schema/beans*](http://www.springframework.org/schema/beans)*"* xmlns:xsi=[*"http://*](http://www.w3.org/2001/XMLSchema-instance)*w*[*ww.w3.org/2001/XMLSchema-instance*](http://www.w3.org/2001/XMLSchema-instance)*"* xmlns:aop=[*"http://*](http://www.springframework.org/schema/aop)*w*[*ww.springframework.org/schema/aop"*](http://www.springframework.org/schema/aop)xsi:schemaLocation=[*"http://*](http://www.springframework.org/schema/beans)*w*[*ww.springframework.org/schema/beans*](http://www.springframework.org/schema/beans)[*http://www.springframework.org/schema/beans/spring-beans.xsd*](http://www.springframework.org/schema/beans/spring-beans.xsd)[*http://www.springframework.org/schema/aop*](http://www.springframework.org/schema/aop) [*http://www.springframework.org/schema/aop/spring-*](http://www.springframework.org/schema/aop/spring-aop.xsd)[*aop.xsd"*](http://www.springframework.org/schema/aop/spring-aop.xsd)>

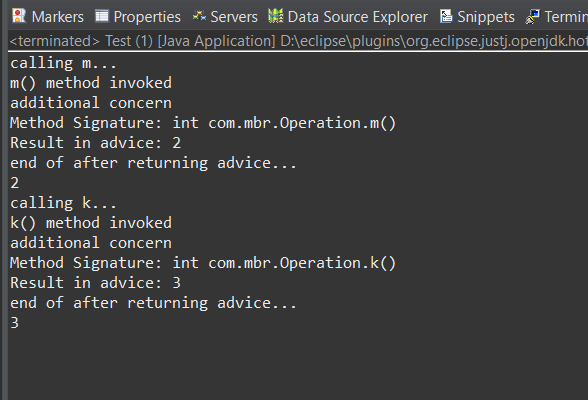
<**bean** id=*"opBean"* class=*"com.mbr.Operation"*> </**bean**>

<**bean** id=*"trackMyBean"* class=*"com.mbr.TrackOperation"*></**bean**>

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

</**beans**>

**Output:**



# Practical N0: 09

## Assignment based Spring JDBC.

### Practical N0: 09A

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

### Employee.java

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

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);

}

}

### Test.java

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();

}

}

}

### applicationContext.xml

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

<beans xmlns=[*"http://*](http://www.springframework.org/schema/beans)*w*[*ww.springframework.org/schema/beans*](http://www.springframework.org/schema/beans)*"* xmlns:xsi=[*"http://*](http://www.w3.org/2001/XMLSchema-instance)*w*[*ww.w3.org/2001/XMLSchema-instance*](http://www.w3.org/2001/XMLSchema-instance)*"* xmlns:p=[*"http://*](http://www.springframework.org/schema/p)*w*[*ww.springframework.org/schema/p"*](http://www.springframework.org/schema/p)

xsi:schemaLocation=[*"http://*](http://www.springframework.org/schema/beans)*w*[*ww.springframework.org/schema/beans*](http://www.springframework.org/schema/beans)[*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd"*](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"*>

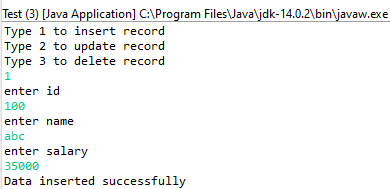
<property name=*"jdbcTemplate"* ref=*"jdbcTemplate"*></property>

</bean>

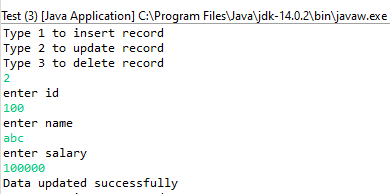
</beans>

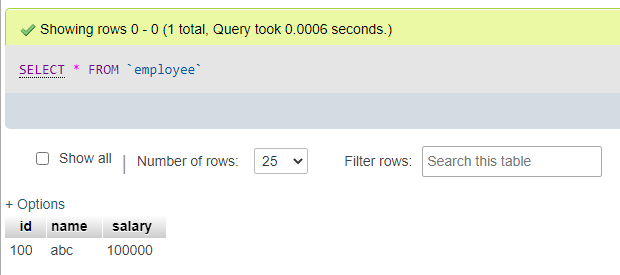
### Output:

**Insert**

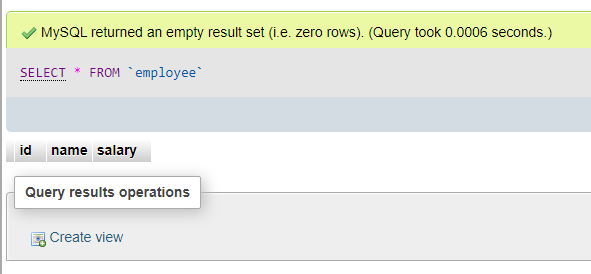
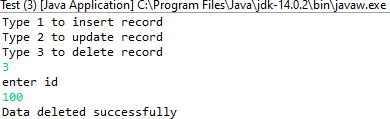


### Update





**Delete**



### Practical N0: 09B

1. **Write a program to demonstrate Prepared Statement in Spring JDBC Template**

### Employee.java

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

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

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");

}

}

### applicationContext.xml

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

<beans xmlns=["http://](http://www.springframework.org/schema/beans)w[ww.springframework.org/schema/beans](http://www.springframework.org/schema/beans)" xmlns:xsi=["http://](http://www.w3.org/2001/XMLSchema-instance)w[ww.w3.org/2001/XMLSchema-instance](http://www.w3.org/2001/XMLSchema-instance)" xmlns:p=["http://](http://www.springframework.org/schema/p)w[ww.springframework.org/schema/p"](http://www.springframework.org/schema/p)

xsi:schemaLocation=["http://](http://www.springframework.org/schema/beans)w[ww.springframework.org/schema/beans](http://www.springframework.org/schema/beans) [http://www.springframework.org/schema/beans/spring-beans-3.0.xsd"](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>

### Output:

