

**LATE BHAUSAHEB HIRAY S.S. TRUST’S INSTITUTE OF COMPUTER APPLICATION**

ISO 9001-2008 CERTIFIED

**S.N. 341, Next to New English School, Govt. Colony, Bandra (East), Mumbai – 400051, Tel: 91-22-26570892/3181**

**Date:13/04/2022**

**CERTIFICATE**

### **This is to certify that Mr. Saurabh Bhausaheb Gawali**

**Roll No. 2021105 is a student of FYMCA Semester-I has completed successfully full-semester practical/assignments of subject Advanced Java Lab for the academic year 2021– 22.**

**Subject In-Charge Director**

**External Examiner**

**LBHSS’s**

**Hiray Institute of Computer Application**

**ADVANCED JAVA PRACTICAL – JOURNAL F.Y.MCA**

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.No** | **Practical Name** | **Date** | **Sign** |
| 1 | **Assignments on Java Generics**  1. Write a Java Program to demonstrate a Generic Class.  2. Write a Java Program to demonstrate Generic Methods.  3. Write a Java Program to demonstrate Wildcards in Java Generics. |  |  |
| 2 | **Assignments on List Interface**  1. Write a Java program to create List containing list of items of type String and use for- --each loop to print the items of the list.  2. Write a Java program to create List containing list of items and use ListIterator interface to print items present in the list. Also print the list in reverse/ backword direction. |  |  |
| 3 | **Assignments on Set Interface**  1. Write a Java program to create a Set containing list of items of type String and print the items in the list using Iterator interface. Also print the list in reverse/ backword direction.  2. Write a Java program using Set interface containing list of items and perform the following operations:  a. Add items in the set.  b. Insert items of one set in to other set.  c. Remove items from the set d. Search the specified item in the set |  |  |
| 4 | **Assignments on Map Interface**  Write a Java program using Map interface containing list of items having keys and associated values and perform the following operations:  a. Add items in the map.  b. Remove items from the map  c. Search specific key from the map  d. Get value of the specified key  e. Insert map elements of one map in to other map.  f. Print all keys and values of the map. |  |  |
| 5 | **Assignments on Lambda Expression**  1. Write a Java program using Lambda Expression to print ”Hello World”.  2. Write a Java program using Lambda Expression with single parameters.  3. Write a Java program using Lambda Expression with multiple parameters to add two numbers.  4. Write a Java program using Lambda Expression to calculate the following:  a. Convert Fahrenheit to Celcius  b. Convert Kilometers to Miles.  5. Write a Java program using Lambda Expression with or without return keyword.  6. Write a Java program using Lambda Expression to concatenate two strings. |  |  |
| 6 | **Assignments based on web application development using JSP**  1. Create a Telephone directory using JSP and store all the information within a database, so that later could be retrieved as per the requirement. Make your own assumptions.  2. Write a JSP page to display the Registration form (Make your own assumptions)  3. Write a JSP program to add, delete and display the records from StudentMaster (RollNo, Name, Semester, Course) table.  4. Design loan calculator using JSP which accepts Period of Time (in years) and Principal Loan Amount. Display the payment amount for each loan and then list the loan balance and interest paid for each payment over the term of the loan for the following time period and interest rate:  a. 1 to 7 year at 5.35%  b. 8 to 15 year at 5.5%  c. 16 to 30 year at 5.75%  5. Write a program using JSP that displays a webpage consisting Application form for change of Study Center which can be filled by any student who wants to change his/ her study center. Make necessary assumptions  6. Write a JSP program that demonstrates the use of JSP declaration, scriptlet, directives, expression, header and footer |  |  |
| 7 | **Assignment based Spring Framework**  1. Write a program to print “Hello World” using spring framework.  2. Write a program to demonstrate dependency injection via setter method.  3. Write a program to demonstrate dependency injection via Constructor. |  |  |
| 8 | **Assignment based Aspect Oriented Programming**  1. Write a program to demonstrate Spring AOP – before advice.  2. Write a program to demonstrate Spring AOP – after advice.  3. Write a program to demonstrate Spring AOP – around advice.  4. Write a program to demonstrate Spring AOP – after returning advice.  5. Write a program to demonstrate Spring AOP – after throwing advice.  6. Write a program to demonstrate Spring AOP – pointcuts. |  |  |
| 9 | **Assignment based Spring JDBC**  1. Write a program to insert, update and delete records from the given table.  2. Write a program to demonstrate PreparedStatement in Spring JdbcTemplate  3. Write a program in Spring JDBC to demonstrate ResultSetExtractor Interface  4. Write a program to demonstrate RowMapper interface to fetch the records from the database |  |  |
| 10 | **Assignment based Spring Boot and RESTful Web Services**  1. Write a program to create a simple Spring Boot application that prints a message.  2. Write a program to demonstrate RESTful Web Services with spring boot |  |  |

**Practical NO: 1.1**

**Aim:** **Write a Java Program to demonstrate a Generic Class.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Program:

**package** practical1;

**public** **class** GenericClassExample <T>{

**private** T dvariable;

**public** T getDvariable() {

**return** dvariable;

}

**public** **void** setDvariable(T dvariable) {

**this**.dvariable = dvariable;

}

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

GenericClassExample<Integer> intvar=**new** GenericClassExample<>(); intvar.setDvariable(10);

System.***out***.println("Variable value="+intvar.getDvariable());

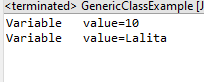
GenericClassExample<String> strvar=**new** GenericClassExample<>(); strvar.setDvariable("Lalita");

System.***out***.println("Variable value="+strvar.getDvariable());

}

}

**Output:**



**Practical No:1.2**

**Aim:** **Write a Java Program to demonstrate Generic Methods**.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Program:**

public class GenericMethod1 {

//print array is generic method

public static <E> void printArray(E[] inputArray)

{

//display array element

for(E element:inputArray)

{

System.out.printf("%s,",element);

}

System.out.println();

}

public static void main(String[] args) {

// TODO Auto-generated method stub Integer [] intArray= {1,2,3,4,5,6};

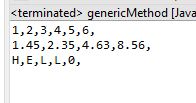
Double [] doubleArray= {1.45,2.35,4.63,8.56}; Character [] charArray= {'H','E','L','L','0'}; printArray(intArray);

printArray(doubleArray); printArray(charArray);

}

}

**Output:**



**Practical NO: 1.3**

**Aim:** **Write a Java Program to demonstrate a Wildcards in Java Generics.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Program:

**package** practical1;

**import** java.util.ArrayList;

**import** java.util.Iterator;

**import** java.util.List;

**public** **class** wildcardExample {

**public** **static** **void** printArray(List<?> mylist)

{

Iterator itr=mylist.iterator();

**while**(itr.hasNext())

{

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

}

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

}

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

ArrayList <Integer>mynumberlist=**new** ArrayList<Integer>();

mynumberlist.add(1);

mynumberlist.add(2);

mynumberlist.add(3);

printArray(mynumberlist);

ArrayList <String>mynamelist=**new** ArrayList<String>();

mynamelist.add("Lalita");

mynamelist.add("lita");

mynamelist.add("Jayesh");

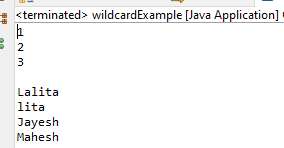
mynamelist.add("Mahesh");

printArray(mynamelist);

}

}

**Output:**



**Practical NO: 2.1**

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

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Program:

**package** practical1;

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** ListInterfaceExample {

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

{

List<String> mylist=**new** ArrayList<String>();

mylist.add("Prakash");

mylist.add("Suresh");

mylist.add("Ramesh");

mylist.add("Jayesh");

**for**(String item: mylist)

{

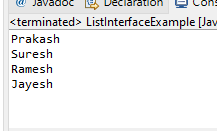
System.***out***.println(item);

}

}

}

**Output:**



**Practical NO: 2.2**

**Aim:** **Write a Java program to create List containing list of items and use ListIterator interface to print items present in the list. Also print the list in reverse/ backword direction.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Program:

**package** practical1;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.Iterator;

**import** java.util.List;

**public** **class** ListInterfaceRevFor

{

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

{

List<Integer> numberlist=**new** ArrayList<Integer>();

**for**(**int** i=1;i<=10;i++)

numberlist.add(i);

// printing the list using iterator

Iterator<Integer> itr=numberlist.iterator();

**while**(itr.hasNext())

{

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

}

// printing the list using iterator in reverse order

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

Collections.*reverse*(numberlist);

Iterator<Integer> itr2=numberlist.iterator();

**while**(itr2.hasNext())

{

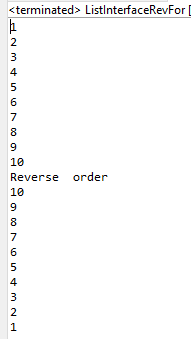
System.***out***.println(itr2.next());

}

}

}

**Output:**



**Practical NO: 3.1**

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

Program: Student.java

**package** com.hirayMca;

**public** **class** Student {

**int** rollno;

String name;

Double percentage;

**public** Student(**int** rollno, String name, Double percentage)

{

**this**.rollno = rollno;

**this**.name = name;

**this**.percentage = percentage;

}

**public** Student()

{

}

**public** **int** getRollno()

{

**return** rollno;

}

**public** **void** setRollno(**int** rollno)

{

**this**.rollno = rollno;

}

**public** String getName()

{

**return** name;

}

**public** **void** setName(String name)

{

**this**.name = name;

}

**public** Double getPercentage()

{

**return** percentage;

}

**public** **void** setPercentage(Double percentage)

{

**this**.percentage = percentage;

}

}

**setExample.java**

**package** com.hirayMca;

**import** java.util.\*;

**public** **class** setExample

{

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

{

String ans=**null**; **int** rn;

String sn; **double** per;

Scanner sc=**new** Scanner(System.***in***); Student s=**new** Student();

**int** choice;

//Creating set of students

HashSet<Student>studentSet=**new** HashSet<Student>();

Iterator i;

//Creating Students

**do**

{

System.***out***.println("Menu");

System.***out***.println("1.Adding student");

System.***out***.println("2.List all students");

System.***out***.println("3.List in reverse direction");

System.***out***.println("4.Exit");

System.***out***.print("Enter your choice(1..4)");

choice=sc.nextInt();

**int** flag=0;

**switch**(choice)

{

**case** 1:

System.***out***.print("Enter rollno");

rn=sc.nextInt();

System.***out***.print("Enter name");

sn=sc.next();

System.***out***.print("Enter percentage");

per=sc.nextDouble();

s=**new** Student(rn,sn,per);

studentSet.add(s);

**break**;

**case** 2:

System.***out***.println("Rollno"+"\t"+"Name"+"\t"+"Percentage");

studentSet.forEach((s1) -> {

System.***out***.println(s1.getRollno()+"\t"+s1.getName()+"\t"+s1.getPercentage());

});

**break**;

**case** 3:

Comparator<Student> c= Comparator.*comparing*(Student::getRollno,Comparator.*reverseOrder*()).thenComparing(Student::getName,Comparator.*reverseOrder*());

List<Student>list=**new** ArrayList<>(studentSet);

list.sort(c);

System.***out***.println("Rollno"+"\t"+"Name"+"\t"+"Percentage");

**for**(Student s2:list)

{

System.***out***.println(s2.getRollno()+"\t"+s2.getName()+"\t"+s2.getPercentage());

}

**break**; **case** 4:

System.*exit*(0);

**break**;

}

//adding student in HashSet

System.***out***.print("Do you wish to continue(y/n)");

ans=sc.next();

}

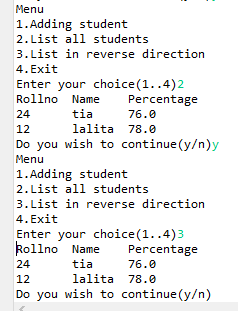
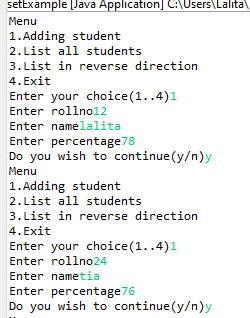
**while**(ans.equals("Y")||ans.equals("y"));

//reteriving the student from set

}

}

**Output:**



**Practical NO: 3.2**

**Aim: Write a Java program using Set interface containing list of items and**

**perform the following operations:**

**a. Add items in the set.**

**b. Insert items of one set in to other set.**

**c. Remove items from the set**

**d. Search the specified item in the set. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Program: Student.java

**package** com.hirayMca;

**public** **class** Student {

**int** rollno;

String name;

Double percentage;

**public** Student(**int** rollno, String name, Double percentage)

{

**this**.rollno = rollno;

**this**.name = name;

**this**.percentage = percentage;

}

**public** Student()

{

}

**public** **int** getRollno()

{

**return** rollno;

}

**public** **void** setRollno(**int** rollno)

{

**this**.rollno = rollno;

}

**public** String getName()

{

**return** name;

}

**public** **void** setName(String name)

{

**this**.name = name;

}

**public** Double getPercentage()

{

**return** percentage;

}

**public** **void** setPercentage(Double percentage)

{

**this**.percentage = percentage;

}

}

**setExample1.java:**

package com.hirayMca;

import java.util.\*;

public class setExample1 {

public static void main(String []args)

{

String ans=null;

int rn;

String sn;

double per;

Scanner sc=new Scanner(System.*in*);

Student s=new Student();

HashSet<Student>studentSet=new HashSet<Student>();

Iterator i;

int choice;

do

{

System.*out*.println("Menu");

System.*out*.println("1.Adding student");

System.*out*.println("2.Removing student");

System.*out*.println("3.Search student");

System.*out*.println("4.List all students");

System.*out*.println("5.Exit");

System.*out*.print("Enter your choice(1..4)");

choice=sc.nextInt();

int flag=0;

switch(choice)

{

case 1:

System.*out*.print("Enter rollno");

rn=sc.nextInt();

System.*out*.print("Enter name");

sn=sc.next();

System.*out*.print("Enter percentage");

per=sc.nextDouble();

s=new Student(rn,sn,per);

studentSet.add(s);

break;

case 2:

System.*out*.print("Enter roll no to remove element");

rn=sc.nextInt();

i=studentSet.iterator();

while(i.hasNext())

{

s=(Student)i.next();

if(s.getRollno()==rn)

{

i.remove();

System.*out*.println("Element is successfully removed");

break;

}

else

{

System.*out*.println("Rollno="+rn);

}

}

break;

case 3:

System.*out*.println("Enter the name of student");

sn=sc.next();

i=studentSet.iterator();

if(!i.hasNext())

{

System.*out*.println("List is empty");

}

while(i.hasNext())

{

s=(Student)i.next();

if(s.getName().equals(sn))

{

System.*out*.println("Record found");

System.*out*.println(s.getRollno()+" "+s.getName()+" "+s.getPercentage()); flag=1;

break;

}

if(flag!=1)

{

System.*out*.println("Record not found search again");

}

}

break;

case 4:

System.*out*.println("Rollno"+"\t"+"Name"+"\t"+"Percentage");

studentSet.forEach((s1) -> {

System.*out*.println(s1.getRollno()+" "+s1.getName()+" "+s1.getPercentage());

});

break;

case 5:

System.*exit*(0);

break;

}

//adding student in HashSet

System.*out*.print("Do you wish to continue(y/n)"); ans=sc.next();

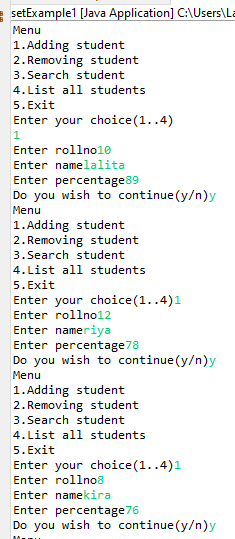
}while(ans.equals("Y")||ans.equals("y"));

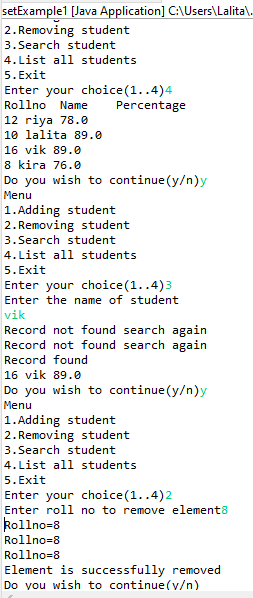
//reteriving the student from set

}

}

**Output:**





**Practical No:4**

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

**a. Add items in the map.**

**b. Remove items from the map.**

**c. Search specific key from the map.**

**d. Get value of the specified key.**

**e. Insert map elements of one map in to other map.**

**f. Print all keys and values of the map.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Program:Book.java:

**package** com.hirayMca;

**public** **class** Book {

**private** **int** id;

**private** String name;

**private** String author;

**public** Book(){ }

**public** Book(**int** id, String name, String author)

{

**this**.id = id;

**this**.name = name;

**this**.author = author;

}

**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** String getAuthor()

{

**return** author;

}

**public** **void** setAuthor(String author)

{

**this**.author = author;

}

}

MapExample.java:

**package** com.hirayMca;

**import** java.util.\*;

**public** **class** MapExample {

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

{

//Creating map of books

Map<Integer,Book>map=**new** HashMap<Integer,Book>();

//Creating books

Book b1=**new** Book(101,"Let us C","Yashwant Kanetkar");

Book b2=**new** Book(102,"Data communication & Networking","Forouzan");

Book b3=**new** Book(103,"Operating System","Achuyut Godbole");

Book b=**new** Book();

//Adding books to map map.put(1,b1);

map.put(2,b2);

map.put(3,b3);

**for**(Map.Entry<Integer, Book> entry:map.entrySet())

{ **int** key=entry.getKey();

b=entry.getValue();

System.***out***.println(b.getId()+" "+b.getName()+" "+b.getAuthor());

}

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

System.***out***.println("Traversing map after removing 2nd element");

**for**(Map.Entry<Integer, Book> entry:map.entrySet())

{

**int** key=entry.getKey(); b=entry.getValue();

System.***out***.println(b.getId()+" "+b.getName()+" "+b.getAuthor());

}

**int** mykey;

Scanner sc=**new** Scanner(System.***in***);

System.***out***.print("Enter the key ");

mykey=sc.nextInt();

**for**(Map.Entry<Integer, Book> entry:map.entrySet())

{ **int** key=entry.getKey();

b=entry.getValue();

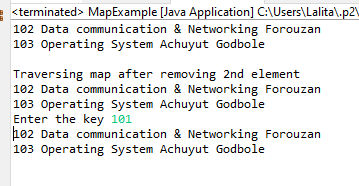
System.***out***.println(b.getId()+" "+b.getName()+" "+b.getAuthor());

}

}

}

**Output:**



**Practical No : 5.1**

**Aim: Write a Java program using Lambda Expression to print ”Hello World”**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Program:**

**interface** HelloWorld {

**void** sayHello();

}

**public** **class** LambdaExpressionExample

{

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

{

HelloWorld helloworld=()->{

System.***out***.println("Hello World");

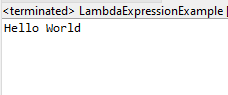
};

helloworld.sayHello();

}

}

**Output:**



**Practical No:5.2**

**Aim:** **Write a Java program using Lambda Expression with single parameters.**

**Program:**

**interface** findCube

{

**int** cube(**int** num);

}

**public** **class** LESingleParameter {

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

{

findCube cubeobj=(**int** x)->{**return** x\*x\*x;};

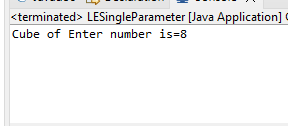
**int** result=cubeobj.cube(2);

System.***out***.println("Cube of Enter number is="+result);

}

}

**Output :**



**Practical NO : 5.3**

**Aim: Write a Java program using Lambda Expression with multiple parameters to add two numbers**.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Program :**

**interface** Calculator

{

**int** add(**int** x,**int** y);

}

**public** **class** LEMultiple

{

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

{

**int** num1=20,num2=10;

Calculator calc=(**int** n1,**int** n2)->{**return** num1+num2;};

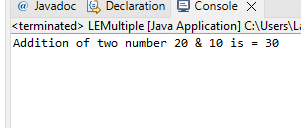
**int** result=calc.add(num1, num2);

System.***out***.println("Addition of two number "+num1+" & "+num2+" is = "+result);

}

}

**Output :**



**Practical No : 5.4**

**Aim : Write a Java program using Lambda Expression to calculate the following:**

**a. Convert Fahrenheit to Celcius b. Convert Kilometers to Miles.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Program : a) Convert Fahrenheit to Celcius**

**import** java.util.function.Function;

**public** **class** FahrenheitToCelcius

{

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

{

Function<Integer,Double>fahrenheittocel= x -> **new** ~~Double~~((x-32)\*5/9);

**double** celcious =fahrenheittocel.apply(100);

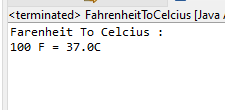
System.***out***.println("Farenheit To Celcius : ");

System.***out***.println("100 F = "+ celcious+"C");

}

}

**Output :**



**b) Convert Kilometers to Miles:**

**import** java.util.function.Function;

**public** **class** KilometerToMile {

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

{

Function<Integer,Double>kilometer= x -> **new** ~~Double~~(x/1.6);

**double** miles = kilometer.apply(10);

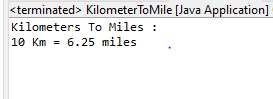
System.***out***.println("Kilometers To Miles : ");

System.***out***.println("10 Km = "+ miles+" miles");

}

}

**Output:**



**Practical No: 5.5**

**Aim:** **Write a Java program using Lambda Expression with or without return keyword.**

**Program:**

**interface** Addable

{

**int** add(**int** a,**int** b);

}

**public** **class** LambdaEWithReturn

{

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

{

//without return keyword

Addable ad1=(a,b)->(a+b);

System.***out***.println(ad1.add(10,20));

//with return keyword

Addable add2=(**int** a,**int** b)->{

**return**(a+b);

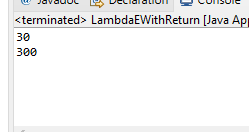
};

System.***out***.println(add2.add(100,200));

}

}

**Output :**



**Practical No : 5.6**

**Aim : Write a Java program using Lambda Expression to concatenate two strings.**

**Program :**

**interface** Constr

{

String addstr(String fname,String lname);

}

**public** **class** ConcatString {

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

{

String fname="Lalita";

String lname="Ingle";

Constr ctr=(String f,String l)->{**return** f+" "+l;};

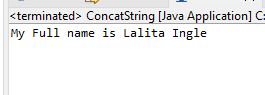
String fullstring=ctr.addstr(fname,lname);

System.***out***.println("My Full name is "+fullstring);

}

}

**Output :**



**Practical No : 6.1**

**Aim :** **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.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Program : Form.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Form</title>

</head>

<body>

<form action="process.jsp">

<table align="center">

<tr align="center">

<td style="color:red;padding:10px">${param.message}</td>

</tr>

</table>

<table border="1" align="center" cellpadding="10px">

<thead>

<tr align="center">

<th colspan="2" style="background-color: yellow; padding: 10px">Telephone-Entry-

Form</th>

</tr>

</thead>

<tbody>

<tr>

<td>Name</td>

<td><input type="text" name="txtName" value="" /></td>

</tr>

<tr>

<td>Telephone Number</td>

<td><input type="text" name="txtTel" value="" /></td>

</tr>

<tr>

<td><input type="submit" value="Add Entry" /></td>

<td><input type="reset" value="Reset" /></td>

</tr>

</tbody>

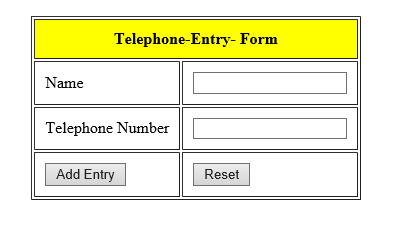
</table>

</form>

</body>

</html>

**Output:**



**Process.jsp:**

<%@page import="java.sql.ResultSet"%>

<%@page import="java.sql.SQLException"%>

<%@page import="java.sql.DriverManager"%>

<%@page import="java.sql.Statement"%>

<%@page import="java.sql.PreparedStatement"%>

<%@page import="java.sql.Connection"%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Process page</title>

</head>

<body>

<%

String uname=request.getParameter("txtName");

String tel=request.getParameter("txtTel");

Connection conn;

PreparedStatement ps;

Statement st;

ResultSet rs;

// connection Initialization

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

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/teldirectory?zeroDateTimeBehavior=convertToNull","root","");

try

{

//checking the record already exist in database or not

st=conn.createStatement();

rs=st.executeQuery("select \* from td");

while(rs.next())

{

if(rs.getString(2).equals(uname)&& rs.getString(3).equals(tel))

{%>

<jsp:forward page="form.jsp">

<jsp:param name="message" value="User is already exits, Give New Entry !!!" />

</jsp:forward>

<%

}

}

//inserting record in database

String sql="insert into td(name,telNo) values(?,?)";

ps=conn.prepareStatement(sql);

ps.setString(1,uname);

ps.setString(2,tel);

ps.executeUpdate();

conn.close();

out.println("<h3 align='center'>");

out.println("Record inserted successfully");

out.println("</h3>");

out.println("<h3 align='center'>");

out.println("Click "+"<a href='form.jsp'>here</a>"+"to Enter another record");

out.println("</h3>");

}

catch(SQLException e)

{

e.printStackTrace();

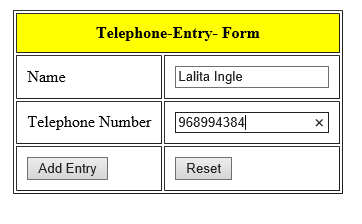
}

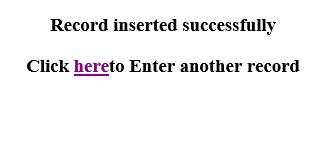
%>

</body>

</html>

**Output:**





**Practical No : 6.2**

**Aim : Write a JSP page to display the Registration form.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Program:**

**Student.jsp:**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<center>

<h2> Student Data </h2>

</center>

<form action="Check.jsp">

<table border="1" align="center" cellpadding="10px" style="font-size: 25px">

<thead>

<tr>

<th colspan="2" style="background-color: pink"> Student Form</th>

</tr>

</thead>

<tbody>

<tr>

<td> Student Name </td>

<td> <input type="text" name="sn" value=""/></td>

</tr>

<tr>

<td> Email</td>

<td> <input type="text" name="em" value=""/></td>

</tr>

<tr>

<td> Date of Birth</td>

<td> <input type="text" name="dob" value=""/></td>

</tr>

<tr>

<td> Education</td>

<td>

<select name="edq">

<option> MCA</option>

<option> MBA</option>

<option> MTech</option>

</select>

</td>

</tr>

<tr>

<td> Language Known</td>

<td>

<input type="checkbox" name="lk" value="Marathi"/>Marathi

<input type="checkbox" name="lk" value="English"/>Hindi

<input type="checkbox" name="lk" value="Hindi"/>English

</td>

</tr>

<tr>

<td> Gender</td>

<td> <input type="radio" name="gen" value="Male"/> Male

<input type="radio" name="gen" value="Female"/>Female

</td>

</tr>

<tr>

<td align="center">

<input type="submit" value="Submit" style="height:40px; width:100px">

</td>

<td align="center">

<input type="reset" value="Reset" style="height:40px; width:100px">

</td>

</tr>

</tbody>

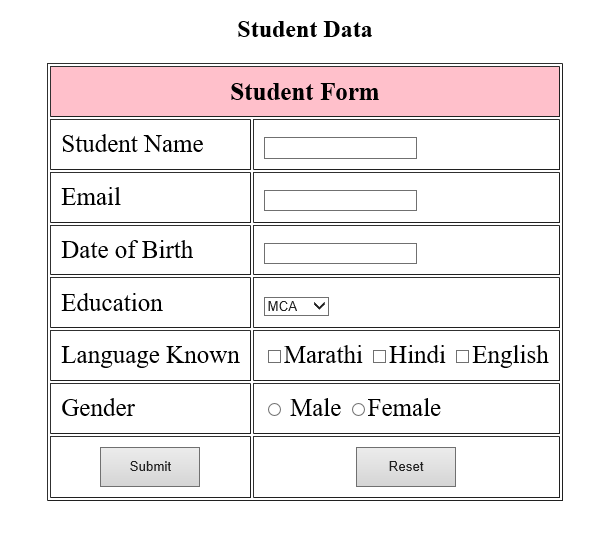
</table>

</form>

</body>

</html>

**Output:**



**Check.jsp:**

<%@page import="java.util.\*" %>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<h2> You are registered successfully</h2>

<%

String name = request.getParameter("sn");

String email = request.getParameter("em");

String dob = request.getParameter("dob");

String edu = request.getParameter("edq");

String lang= request.getParameter("Hindi");

String gender = request.getParameter("gen");

String[] langk=request.getParameterValues("lk");

List lst= Arrays.asList("langk");

request.setAttribute("langk",lst);

List<String>lk=(List<String>)request.getAttribute("langk");

out.println("<table>");

out.println("<tr>");

out.println("<td>");

out.println("Student Name");

out.println("</td>");

out.println("<td>");

out.println(name);

out.println("</td>");

out.println("</tr>");

out.println("<tr>");

out.println("<td>");

out.println("Email ");

out.println("</td>");

out.println("<td>");

out.println(email);

out.println("</td>");

out.println("</tr>");

out.println("<tr>");

out.println("<td>");

out.println("Date of Birth");

out.println("</td>");

out.println("<td>");

out.println(dob);

out.println("</td>");

out.println("</tr>");

out.println("<tr>");

out.println("<td>");

out.println("Educational Qulification");

out.println("</td>");

out.println("<td>");

out.println(edu);

out.println("</td>");

out.println("</tr>");

out.println("<tr>");

out.println("<td>");

out.println("Language Known");

out.println("</td>");

out.println("<td>");

for(String item:lk)

{

out.println(item);

}

out.println("</td>");

out.println("</tr>");

out.println("<tr>");

out.println("<td>");

out.println("Gender");

out.println("</td>");

out.println("<td>");

out.println(gender);

out.println("</td>");

out.println("</tr>");

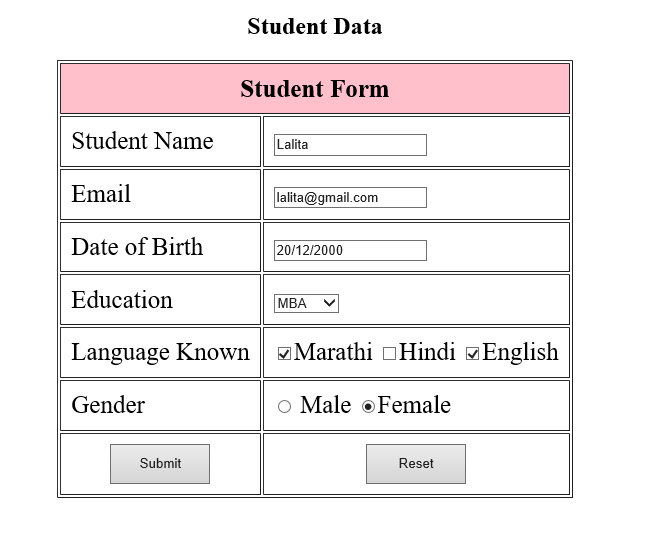
out.println("</table>");

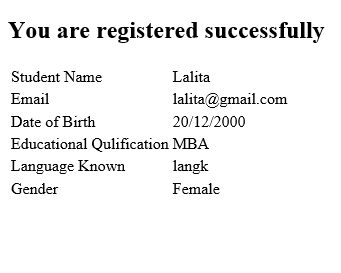
%>

</body>

</html>

**Output:**





**Practical No : 6.3**

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

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Program :**

##### **student.jsp code:**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<form action="insert.jsp">

<table>

<tr>

<td> Student Name</td>

<td> <input type="text" name="sname" id="sname" />

</td>

</tr>

<tr>

<td> Course</td>

<td> <input type="text" name="course" id="course" />

</td>

</tr>

<tr>

<td> Semester</td>

<td> <input type="text" name="sem" id="sem"/></td>

</tr>

<tr>

<td> <input type="submit" id="submit" value="Submit" name="submit"/></td>

<td> <input type="reset" id="reset" value="Reset" name="reset"/></td>

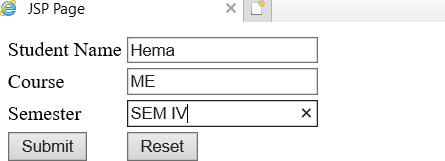
</tr>

</table>

</form>

</body

</html>



##### **insert.jsp code:**

<%@page import="java.sql.Connection"%>

<%@page import="java.sql.\*"%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<%

String name = request.getParameter("sname"); String crs = request.getParameter("course"); String sem = request.getParameter("sem"); Connection conn;

PreparedStatement ps; try

{

Class.forName("com.mysql.jdbc.Driver"); conn =

DriverManager.getConnection("jdbc:mysql://localhost:3306/stab?zeroDateTimeBehavior=convertT oNull","root","");

String sql = "insert into stab(stname,stcource,stsame)value(?,?,?)"; ps = conn.prepareStatement(sql);

ps.setString(1, name); ps.setString(2, crs); ps.setString(3, semt); ps.executeUpdate();

conn.close();

}

catch(Exception e)

{

e.printStackTrace();

}

%>

<jsp:forward page="display.jsp"/>

</body>

</html>

##### **update.jsp code:**

<%@page import="java.sql.\*;" %>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%

if(request.getParameter("submit")!=null)

{

String id=request.getParameter("id");

String name = request.getParameter("sname"); String crs = request.getParameter("course"); String semt = request.getParameter("sem");

Connection conn; PreparedStatement ps;

Class.forName("com.mysql.jdbc.Driver"); conn =

DriverManager.getConnection("jdbc:mysql://localhost:3306/fuser?zeroDateTimeBehavior=convert ToNull","root","");

String sql = "update stab set stname=?, stcource=?, stsame=? where id=?"; ps = conn.prepareStatement(sql);

ps.setString(1, name); ps.setString(2, crs); ps.setString(3, semt); ps.setString(4, id);

ps.executeUpdate(); out.println("Updated Successfully!!!");

}

%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<h2> Student Update</h2>

<form method="post">

<%

Connection conn; Statement stmt;

ResultSet rs; PreparedStatement ps;

Class.forName("com.mysql.jdbc.Driver"); conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/FYMCA?zeroDateTimeBehavior

=convertToNull","root","");

String id = request.getParameter("id");

ps = conn.prepareStatement("select \* from stab where id = ?"); ps.setString(1, id);

rs= ps.executeQuery();

while(rs.next())

{

%>

<table>

<tr>

<td> Student Name</td>

<td> <input type="text" name="sname" id="sname" value="<%=rs.getString("stname")

%>"/></td>

</tr>

<tr>

<td> Course</td>

<td> <input type="text" name="course" id="course" value="<%=rs.getString("stcource")

%>" /></td>

</tr>

<tr>

<td> Semester</td>

<td> <input type="text" name="sem" id="sem" value="<%=rs.getString("stsame")

%>"/></td>

</tr>

<%

}

%>

<tr>

<td> <input type="submit" id="submit" value="Submit" name="submit"/></td>

<td> <input type="reset" id="reset" value="Reset" name="reset"/></td>

<td>

<a href="display.jsp"> Home</a>

</td>

</tr>

</table>

</form>

</body>

</html>

##### delete.jsp code:

<%@page import="java.sql.\*;" %>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%

String id=request.getParameter("id");

Connection conn; PreparedStatement ps;

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

=DriverManager.getConnection("jdbc:mysql://localhost:3306/FYMCA?zeroDateTimeBehavior=con vertToNull","root","");

String sql="delete from stab where id=?"; ps=conn.prepareStatement(sql);

ps.setString(1, id); ps.executeUpdate();

out.println("Record deleted successfully!!!");

%>

<a href="display.jsp">Home</a>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

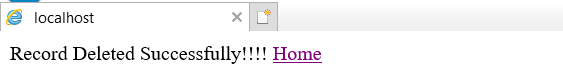
</head>

<body>

<h1>delete page</h1>

</body>

</html>



##### **display.jsp code:**

<%@page import="java.sql.Connection"%>

<%@page import="java.sql.\*"%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<table border="1" cellpadding="15px">

<thead style="font-size: 20px" >

<tr>

<td> Name</td>

<td> Course</td>

<td> Semester</td>

<td> Edit </td>

<td> Delete </td>

</tr>

</thead>

<tbody>

<%

Connection conn; Statement stmt; ResultSet rs;

try

{

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

conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/FYMCA?zeroDateTimeBehavior

=convertToNull","root","");

stmt=conn.createStatement(); rs=stmt.executeQuery("select \* from stab");

while(rs.next())

{

%>

%>

</tbody>

String id = rs.getString("id");

<tr>

<td><%=rs.getString(2)%></td>

<td><%=rs.getString(3)%></td>

<td><%=rs.getString(4)%></td>

<td><a href="update.jsp?id=<%=id%>"> Edit </a> </td>

<td><a href="delete.jsp?id=<%=id%>"> Delete </a></td>

</tr>

<%

}

}

catch(Exception e)

{

e.printStackTrace();

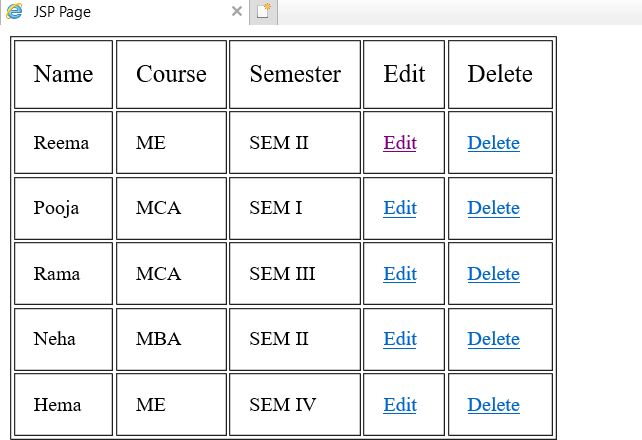
}

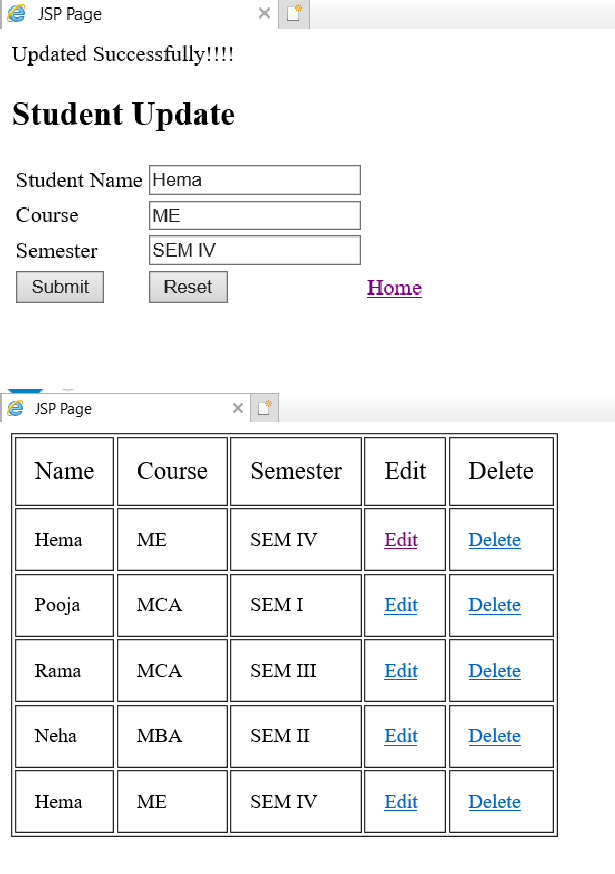
</table>

</body>

</html>

**Output:**





**Practical No : 6.4**

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

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

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

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

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Program :**

**LoanCalculator.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Loan Calculator design Page</title>

</head>

<body>

<form action="calculator.jsp" method="post">

Enter Loan Amount:<input type="text" name="LoanAmount">

<br><br>

Enter No of year:<input type="text" name="year">

<br><br>

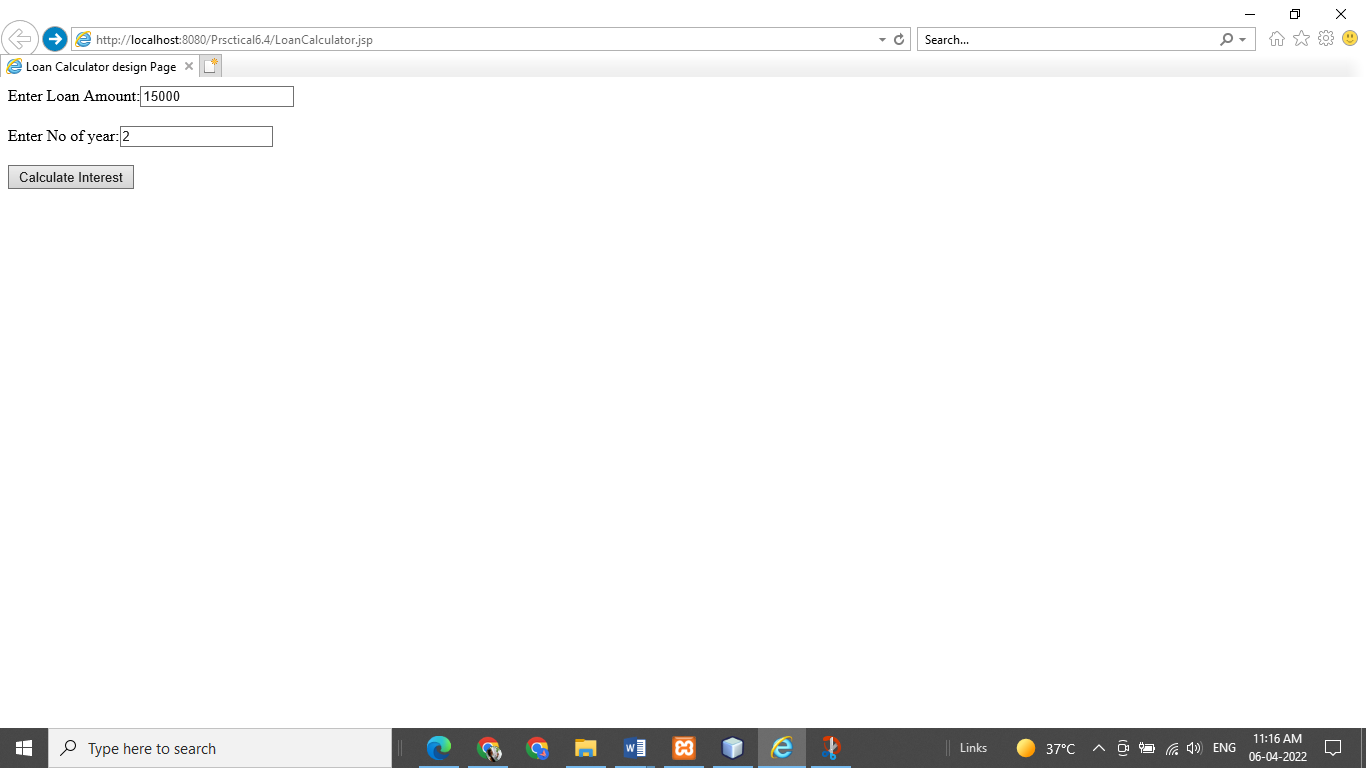
<input type="submit" value="Calculate Interest">

</form>

</body>

</body>

</html>

**Output:**

**Calculator.jsp:**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>calculator code page</title>

</head>

<body>

<%@include file="LoanCalculator.jsp" %>

<%

double loanamount = Double.parseDouble(request.getParameter("LoanAmount"));

double years = Double.parseDouble(request.getParameter("year"));

double interest = 0.0;

if(years>=1&& years<=7)

{

interest=(loanamount\*5.35)/100; //loanamount\*5.35%

out.println("Interest to paid per month:"+ interest);

}

else if(years>=8 && years<=15)

{

interest=(loanamount\*5.50)/100; //loanamount\*5.50%

out.println("Interest to paid per month:"+ interest);

}

else if(years>=16 && years<=30)

{

interest=(loanamount\*5.75)/100; //loanamount\*5.75%

out.println("Interest to paid per month:"+ interest);

}

else

{

out.println("Invalid Year");

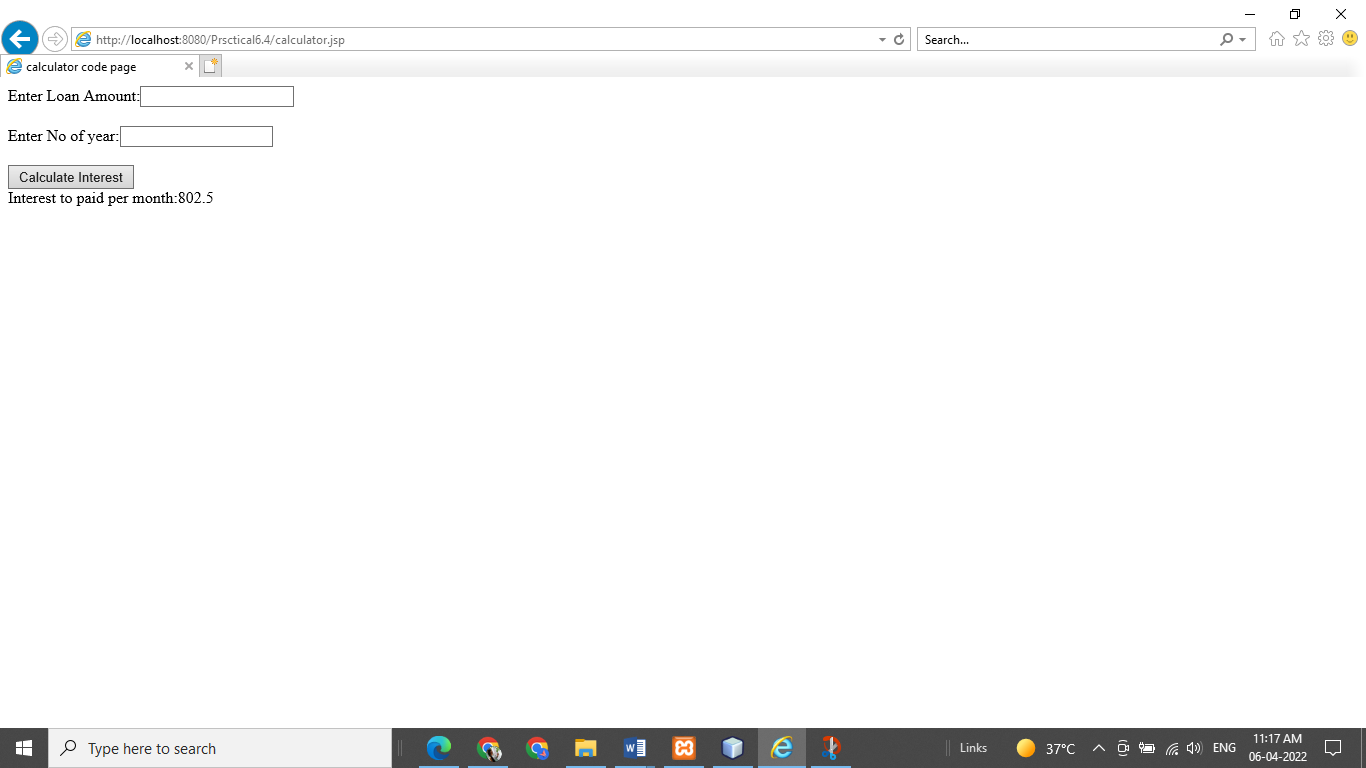
}

%>

</body>

</html>

**Output :**



**Practical No :6.5**

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

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Program:**

**Form.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Practical</title>

</head>

<body>

<h1 align="Center">Application form for Change of Study Center</h1>

<form action="process.jsp">

<table border="1" align="center" cellpadding="20px">

<tbody>

<tr>

<td>Registration id</td>

<td><input type="text" name="txtRegid" value="" /></td>

</tr>

<tr>

<td>Name</td>

<td><input type="text" name="txtName" value="" /></td>

</tr>

<tr>

<td>Old Study center address</td>

<td><textarea name="txtOldaddress" rows="4" cols="20">

</textarea></td>

</tr>

<tr>

<td>New Study center address</td>

<td><textarea name="txtNewaddress" rows="4" cols="20">

</textarea></td>

</tr>

<tr>

<td><input type="submit" value="Change It" /></td>

<td><input type="reset" value="Reset" /></td>

</tr>

</tbody>

</table>

</form>

</body>

</html>

**Process.jsp:**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<h1 align="center">Address is Successfully change</h1>

<%

String regid=request.getParameter("txtRegid");

String name=request.getParameter("txtName");

String oldadd=request.getParameter("txtOldaddress");

String newadd=request.getParameter("txtNewaddress");

out.println("<table border='1' align='center' cellpadding='20px' ");

out.println("<tr>");

out.println("<td>");

out.println("Registration Id");

out.println("</td>");

out.println("<td>");

out.println(regid);

out.println("</td>");

out.println("</tr>");

out.println("<tr>");

out.println("<td>");

out.println("Name");

out.println("</td>");

out.println("<td>");

out.println(name);

out.println("</td>");

out.println("</tr>");

out.println("<tr>");

out.println("<td>");

out.println("Old address");

out.println("</td>");

out.println("<td>");

out.println(oldadd);

out.println("</td>");

out.println("</tr>");

out.println("<tr>");

out.println("<td>");

out.println("New address");

out.println("</td>");

out.println("<td>");

out.println(newadd);

out.println("</td>");

out.println("</tr>");

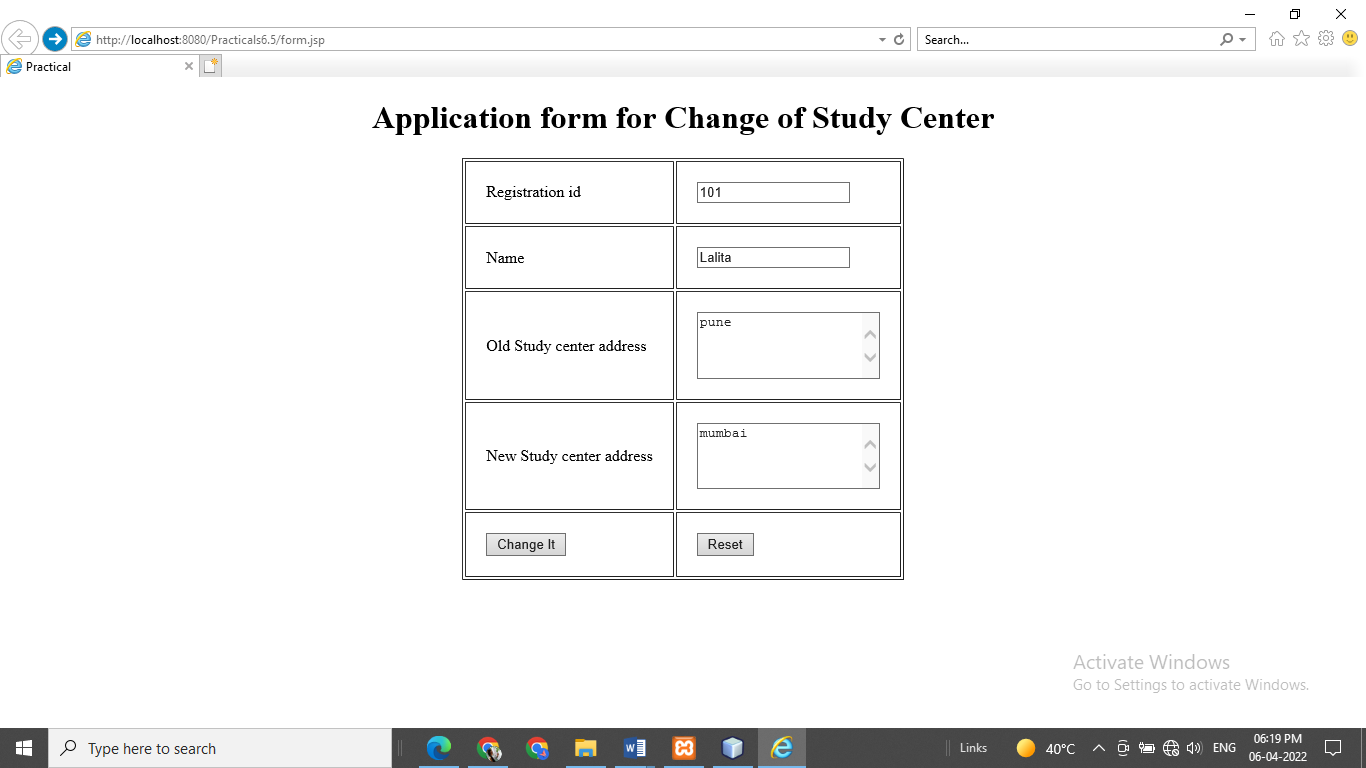
out.println("</table>");

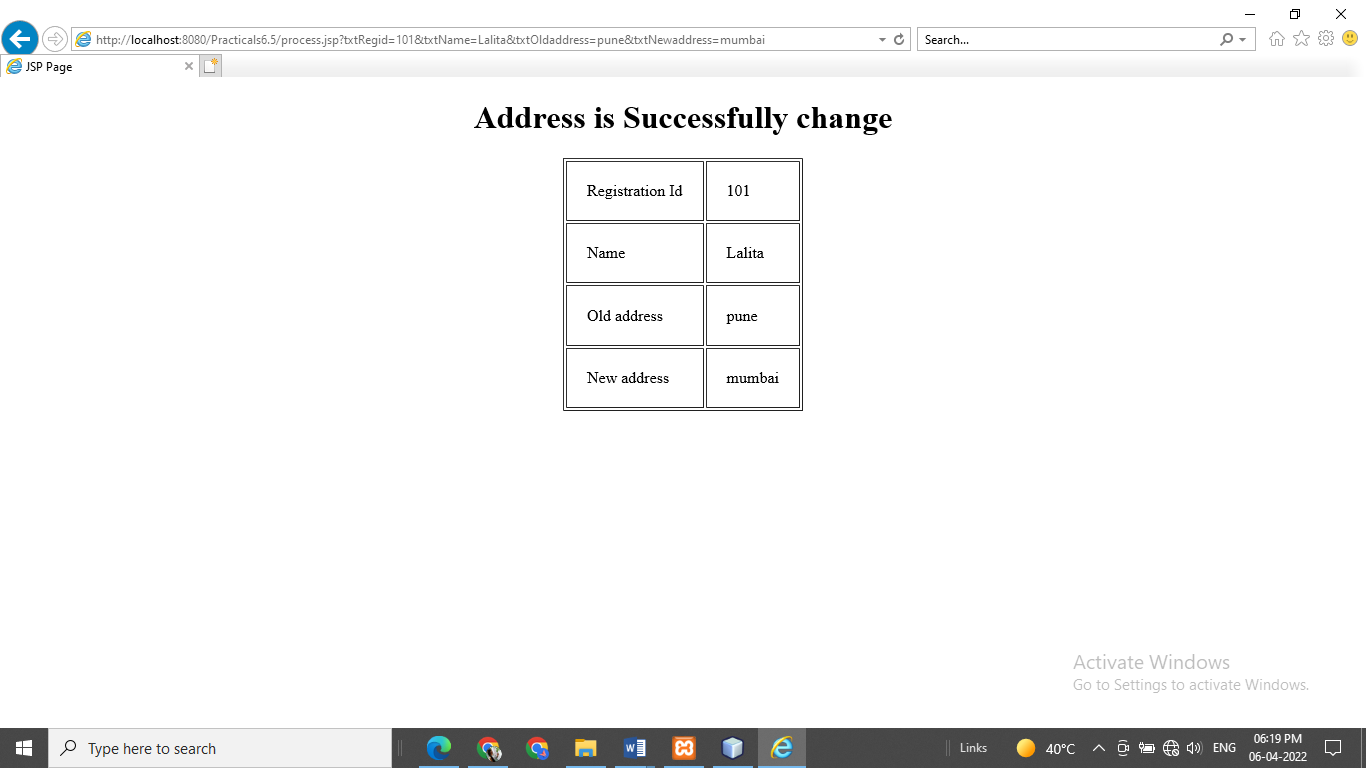
%>

</body>

</html>

**Output:**





**Practical No:6.6**

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

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Program : index.jsp:**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>index Page</title>

<style>

#content

{

margin-top:10px;

}

</style>

</head>

<body>

<%@include file="header.jsp" %>

<div id="content">

<%! int number1=10,number2=20;%>

Value of number is= <%= number1%> <br>

Today's Date is <% out.println(new java.util.Date()); %><br>

Number1=<%=number1%><br> Number2=<%=number2%><br>

Addition of two numbers are <%=number1+number2%><br>

</div>

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

</body>

</html>

**Header.jsp:**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>header Page</title>

<style>

#menu

{

background-color:orange; padding: 2px;

}

li

{

display:inline; padding:5px;

}

a

{

text-decoration: none; color:brown;

}

</style>

</head>

<body>

<div id="menu">

<ul>

<li> <a href="#">Home</a></li>

<li> <a href="#">Service</a></li>

<li> <a href="#">Contact us </a></li>

<li> <a href="#">About us</a></li>

</ul>

</div>

</body>

</html>

**Footer.jsp:**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>footer Page</title>

<style>

#foot

{

color:white;

background-color: black; margin-top:200px; padding:10px;

}

</style>

</head>

<body>

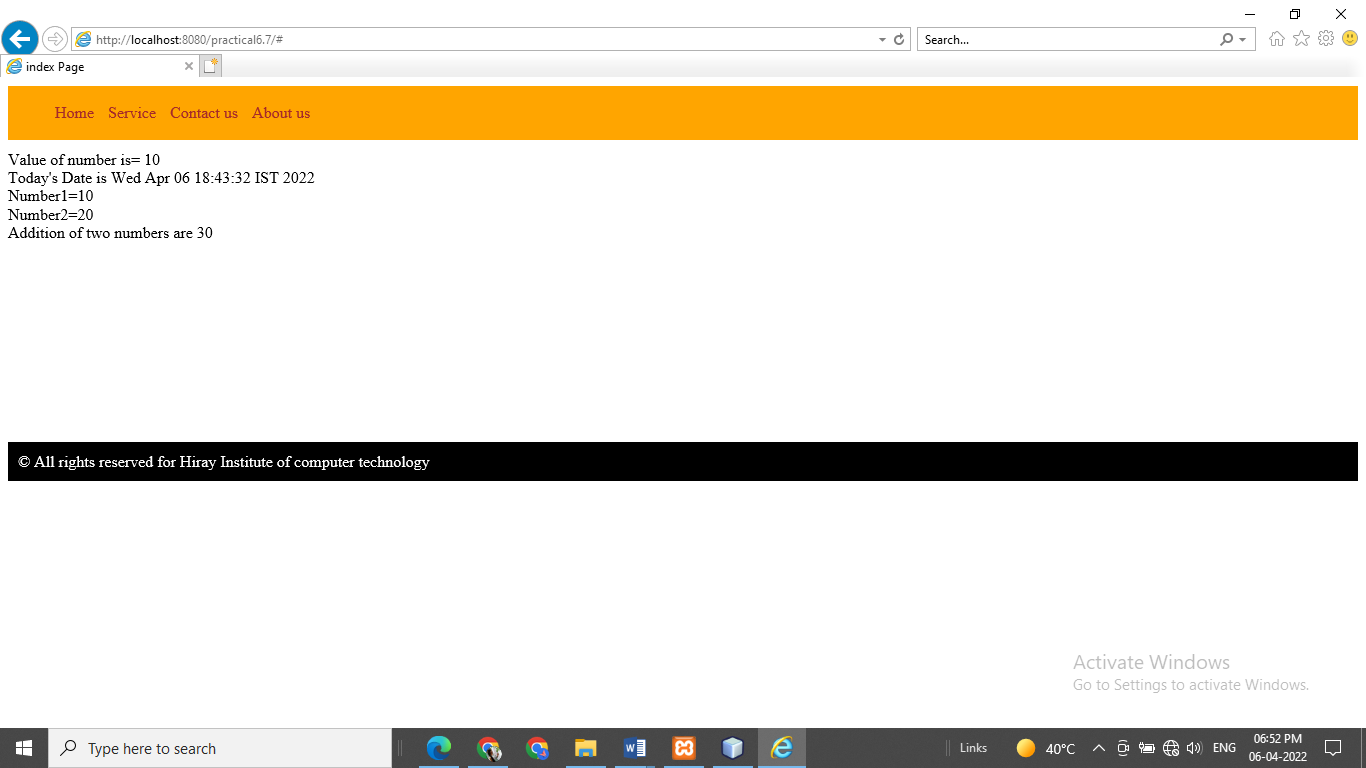
<div id="foot">

&COPY; All rights reserved for Hiray Institute of computer technology</div>

</body>

</html>

**Output:**



**Practical No : 7.1**

**Aim : Write a program to print “Hello World” using spring framework.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Program:**

**HelloWorld.java:**

package com.spring;

public class HelloWorld

{

public void display()

{

System.out.println("Hello World!");

}

}

**Config.xml :**

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

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

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

<bean id="helloworld" class="com.hiraymca.HelloWorld" />

</beans>

**App.java**:

package com.spring;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class App {

public static void main(String[] args)

{

ApplicationContext context = newClassPathXmlApplicationContext("config.xml"); HelloWorld hw = (HelloWorld) context.getBean("helloworld");hw.display();

}

}

**Output : Hello World**

**Practical No:7.2**

**Aim : Write a program to demonstrate dependency injection via setter method.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Program:Employee.java**

**package** com.hiraymca;

**public class** Employee { **private** String ename; **private int** eage; **private int** esal;

**public** String getEname() {

**return** ename;

}

**public void** setEname(String ename) {

**this**.ename = ename;

}

**public int** getEage() {

**return** eage;

}

**public void** setEage(**int** eage) {

**this**.eage = eage;

}

**public int** getEsal() {

**return** esal;

}

**public void** setEsal(**int** esal) {

**this**.esal = esal;

}

**public void** display()

{

System.***out***.println("Name="+ename+"\n"+"Age="+eage+"\n"+"Salary="+ esal);

}

}

**MainApp.java:**

**package** com.hiraymca;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**public class** MainApp {

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

// **TODO** Auto-generated method stub ApplicationContext context=**new**

ClassPathXmlApplicationContext("Beans.xml");

Employee e=(Employee)context.getBean("employee"); e.display();

}

}

**Bean.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)*"* xsi:schemaLocation=*"*[*http://www.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)*"*>

<bean id=*"employee"* class=*"com.hiraymca.Employee"*>

<property name=*"ename"* value=*"Lalita"*></property>

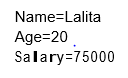
<property name=*"eage"* value=*"20"*> </property>

<property name=*"esal"* value=*"75000"*></property>

</bean>

</bean>

**Output:**



**Practical No : 7.3**

**Aim : Write a program to demonstrate dependency injection via Constructor.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Program:**

**Employee.java:**

package com.hiraymca;

public class Employee

{

private String ename;

private int eage;

private int esal;

public Employee(String ename, int eage, int esal)

{

super();

this.ename = ename;

this.eage = eage;

this.esal = esal;

}

public Employee()

{

this.ename="xyz";

this.eage=75;

this.esal=5000;

}

public String getEname() { return ename; }

public void setEname(String ename) {this.ename = ename;}

public int getEage() {return eage;}

public void setEage(int eage) {this.eage = eage;}

public int getEsal() {return esal;}

public void setEsal(int esal) { this.esal = esal;}

public void display()

{

System.out.println("Name="+ename);

System.out.println("Age="+eage);

System.out.println("Salary="+esal);

}

}

**MainAap.java:**

package com.hiraymca;

import org.springframework.beans.factory.BeanFactory;

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

public static void main(String[] args) {

// TODO Auto-generated method stub

BeanFactory factory=new ClassPathXmlApplicationContext("Beans.xml"); Employee e=(Employee)factory.getBean("emp");

e.display();

}

}

**Bean.xml:**

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

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

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

<bean id="emp" class="com.hiraymca.Employee" >

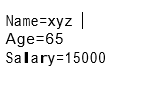
<constructor-arg index="0" value="sukhiram"/>

<constructor-arg index="1" value="65"/>

<constructor-arg index="2" value="15000"/>

</bean>

**Output:**



**Practical No :8.1**

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

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Program:**

**Student.java**

package aop.abprogram;

public class student {

private int id;

private String name;

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 void display() {

System.out.println("Id="+id);

System.out.println("Name="+name);

}

}

**Logging.java:**

package aop.abprogram;

public class Logging {

public void beforeAdvice() {

System.out.println("Student data will be printed soon");

}

public void afterAdvice() {

System.out.println("Student data is displayed successfully");

}

}

**Config.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"

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

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

http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop-3.0.xsd ">

<bean id="student" class="aop.abprogram.student">

<property name="id" value="87"></property>

<property name="name" value="Lalita"></property>

</bean>

<bean id="logging" class="aop.abprogram.Logging">

</bean>

<aop:aspectj-autoproxy></aop:aspectj-autoproxy>

<aop:config>

<aop:aspect id="logaspect" ref="logging">

<aop:before method="beforeAdvice" pointcut="execution(public void

display())"/>

display())"/>

<aop:after method="afterAdvice" pointcut="execution(public void

</aop:aspect>

</aop:config>

</beans>

**App.java:**

package aop.abprogram;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class App {

public static void main( String[] args )

{

ApplicationContext context = new ClassPathXmlApplicationContext("config.xml");

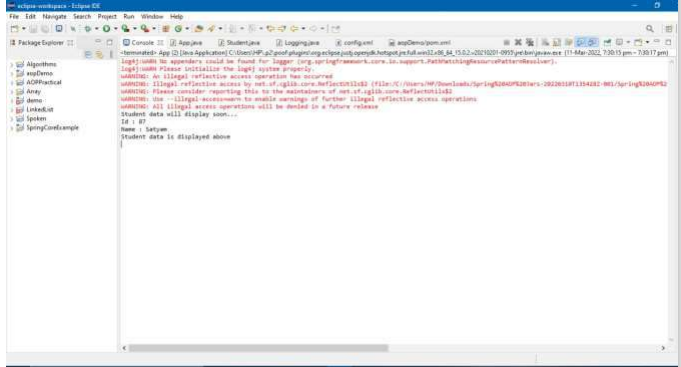
student s=(student) context.getBean("student");

s.display();

}

}

Output:



**Practical No : 8.2**

**Aim :** **Write a program to demonstrate Spring AOP – after advice.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Program :**

**Logging.java :**

package com.hiraymca;

public class Logging {

public void afterAdvice()

{

System.out.println("Student Profile done ");

}

}

**Student.java**

package com.hiraymca;

public class Student {

private int age;

private String name;

public int getAge() {

System.out.println("Age-"+age);

return age;}

public void setAge(int age) {

this.age = age;

}

public String getName() {

System.out.println("Name="+name);

return name;

}

public void setName(String name) {

this.name = name;

}

}

**MainApp.java**

package com.hiraymca;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp

{

public static void main(String[] args) {

stub ApplicationContext context=new

ClassPathXmlApplicationContext("Beans.xml");

Student s=(Student)context.getBean("student");

s.getName();

s.getAge();

}

}

**Beans.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:aop="http://www.springframework.org/schema/aop" xmlns:c="http://www.springframework.org/schema/c" xmlns:lang="http://www.springframework.org/schema/lang" xmlns:util="http://www.springframework.org/schema/util" xsi:schemaLocation="http://www.springframework.org/schema/beans

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

http://www.springframework.org/schema/lang/spring-lang-4.3.xsd http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop-4.3.xsd http://www.springframework.org/schema/util

http://www.springframework.org/schema/util/spring-util-4.3.xsd">

<aop:config>

<aop:aspect id="log" ref="logging">

<aop:pointcut expression="execution(\* com.hiraymca.\*.\*(..))" id="myid"/>

<aop:after method="afterAdvice" pointcut-ref="myid" />

</aop:aspect>

</aop:config>

<bean id="student" class="com.hiraymca.Student">

<property name="name" value="Sukhiram"></property>

<property name="age" value="35"></property>

</bean>

<bean id="logging" class="com.hiraymca.Logging"></bean>

</beans>

**Output**

Name=Sukhiram

Student Profile done

Age-35

Student Profile done

**Practical NO :8.3**

**Aim :** **Write a program to demonstrate Spring AOP – around advice**.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Code:(pom.xml(where all dependencies are loaded))**

<project xmlns="http://maven.apache.org/POM/4.0.0"

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

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>aop</groupId>

<artifactId>practical4prog\_2</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>practical4prog\_2</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>4.2.4.RELEASE</version>

</dependency>

<!-- AspectJ dependencies -->

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjrt</artifactId>

<version>1.7.4</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjtools</artifactId>

<version>1.7.4</version>

</dependency>

</dependencies>

</project>

**● Code:(Student.java)**

package aop.practical4prog\_2;

public class Student {

private int id;

private String name;

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 String display()

{

System.out.println("Id:"+id);

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

return "Student data was displayed successfully";

}

public void validate(int age)

{

if (age<18)

{

throw new ArithmeticException();

}

else

{

System.out.println("Valid age")

}

}

}

**● Code:(Logging.java)**

package aop.practical4prog\_2;

import org.aspectj.lang.ProceedingJoinPoint;

public class Logging {

public void aroundAdvice(ProceedingJoinPoint p) throws Throwable

{

System.out.println("Target display method will display student data soon...");

}

p.proceed();//call the target method

i.e display()

System.out.println("Student data is displayed");

}

**● Code:(config.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"

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

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

http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop-3.0.xsd ">

<bean id="student" class="aop.practical4prog\_2.Student">

<property name="id" value="87"></property>

<property name="name" value="Lalita"></property>

</bean>

<bean id="logging" class="aop.practical4prog\_2.Logging">

</bean>

● Code:(Logging.java)

package aop.practical4prog\_2;

import org.aspectj.lang.ProceedingJoinPoint;

public class Logging {

public void aroundAdvice(ProceedingJoinPoint p) throws Throwable

{

soon...");

}

System.out.println("Target display method will display student data

p.proceed();//call the target method

i.e display()

System.out.println("Student data is

displayed");

}

● Code:(config.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"

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

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

http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop-3.0.xsd ">

<bean id="student" class="aop.practical4prog\_2.Student">

<property name="id" value="87"></property>

<property name="name" value="Lalita"></property>

</bean><aop:aspectj-autoproxy></aop:aspectj-autoproxy>

<aop:config>

<aop:aspect id="logaspect" ref="logging">

<aop:around method="aroundAdvice" pointcut="execution(public void display())"/>

</aop:aspect>

</aop:config>

</beans>

**● Code:(App.java)**

package aop.practical4prog\_2;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class App

{

public static void main( String[] args )

{

ApplicationContext context = new

ClassPathXmlApplicationContext("config.xml");

Student s =(Student) context.getBean("student");

s.display();

//s.validate(19);

}

}

**● Output:**

Student data will display soon..

Id= 87

Name= Lalita

Student data is displayed above

**Practical No: 8.4 & 8.5**

**Aim: Write a program to demonstrate Spring AOP – after returning advice.**

**Aim : Write a program to demonstrate Spring AOP – after throwing advice**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**● Code:(pom.xml(All dependencies are loaded here))**

<project xmlns="http://maven.apache.org/POM/4.0.0"

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

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>aop</groupId>

<artifactId>practical4prog\_2</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>practical4prog\_2</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>4.2.4.RELEASE</version>

</dependency>

<!-- AspectJ dependencies -->

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjrt</artifactId>

<version>1.7.4</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjrt</artifactId>

<version>1.8.5</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjtools</artifactId>

<version>1.7.4</version>

</dependency>

</dependencies>

</project>

**● Code:(Student.java)**

package aop.practical4prog\_2;

public class Student {

private int id;

private String name;

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 String display()

{

System.out.println("Id:"+id);

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

return "Student data is printed";

}

public void validate(int age)

{

if (age<18)

{

throw new ArithmeticException();

}

else{

System.out.println("Valid age");

}

}

}

**● Code:(Logging.java)**

package aop.practical4prog\_2;

import org.aspectj.lang.ProceedingJoinPoint;

public class Logging {

public void aroundAdvice(ProceedingJoinPoint p) throws Throwable

{

System.out.println("Target display method will display student data soon...");

}

p.proceed();//call the target method

i.e display()

System.out.println("Student data is displayed");

public void afterReturning(Object retVal)

{

System.out.println(retVal);

}

public void afterThrowing(Exception error)

{

System.out.println("Some exception occured");

System.out.println(error);

}

}

**● Code:(config.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"

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

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

http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop-3.0.xsd ">

<bean id="student" class="aop.practical4prog\_2.Student">

<property name="id" value="87"></property>

<property name="name" value="Lalita"></property>

</bean>

<bean id="logging" class="aop.practical4prog\_2.Logging">

</bean>

<aop:aspectj-autoproxy></aop:aspectj-autoproxy>

<!-- <aop:config> -->

<!-- <aop:aspect id="logaspect" ref="logging"> -->

<!-- <aop:around method="aroundAdvice" pointcut="execution(public void display())"/>

-->

<!-- </aop:aspect> -->

<!-- </aop:config> -->

<aop:config>

<aop:aspect id="logaspect" ref="logging">

<aop:after-returning method="afterReturning" returning="retVal"

pointcut="execution(public String display())"/>

<aop:after-throwing method="afterThrowing" throwing="error"

pointcut="execution(public void validate())"/>

</aop:aspect>

</aop:config>

</beans>

**● Code:(App.java)**

package aop.practical4prog\_2;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class App

{

public static void main( String[] args )

{

ApplicationContext context = new

ClassPathXmlApplicationContext("config.xml");

Student s =(Student) context.getBean("student");

s.display();

s.validate(19);

}

}

**Output:**

Id : 87

Name : Lalita

Method was successfully executed

Valid age

**Practical No : 8.6**

**Aim : Write a program to demonstrate Spring AOP – pointcuts**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**● Code:(pom.xml)**

<project xmlns="http://maven.apache.org/POM/4.0.0"

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

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>aop</groupId>

<artifactId>practical4prog\_2</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>practical4prog\_2</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>4.2.4.RELEASE</version>

</dependency>

<!-- AspectJ dependencies -->

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjrt</artifactId>

<version>1.7.4</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjrt</artifactId>

<version>1.8.5</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjtools</artifactId>

<version>1.7.4</version>

</dependency>

</dependencies>

</project>

**● Code:(Student.java)**

package aop.practical4prog\_2;

public class Student {

private int id;

private String name;

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 String display()

{

System.out.println("Id:"+id);

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

return "Student data is printed";

}

public void validate(int age)

{

if (age<18)

{

throw new ArithmeticException();

}

else

{

System.out.println("Valid age");

}

}

}

**● Code:(Logging.java)**

package aop.practical4prog\_2;

import org.aspectj.lang.ProceedingJoinPoint;

public class Logging {

public void aroundAdvice(ProceedingJoinPoint p) throws Throwable

{

System.out.println("Target display method will display student data soon...");

}

p.proceed();//call the target method

i.e display()

System.out.println("Student data is displayed");

public void afterReturning(Object retVal)

{

System.out.println(retVal);

}

public void afterThrowing(Exception error)

{

System.out.println("Some exception occured");

System.out.println(error);

}

}

**● Code:(config.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"

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

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

http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop-3.0.xsd ">

<bean id="student" class="aop.practical4prog\_2.Student">

<property name="id" value="87"></property>

<property name="name" value=”Lalita"></property>

</bean>

<bean id="logging" class="aop.practical4prog\_2.Logging">

</bean>

<aop:aspectj-autoproxy></aop:aspectj-autoproxy>

<!-- <aop:config> -->

<!-- <aop:aspect id="logaspect" ref="logging"> -->

<!-- <aop:around method="aroundAdvice" pointcut="execution(public void display())"/>

-->

<!-- </aop:aspect> -->

<!-- </aop:config> -->

<aop:config>

<aop:aspect id="logaspect" ref="logging">

<aop:after-returning method="afterReturning" returning="retVal"

pointcut="execution(public String display())"/>

<aop:after-throwing method="afterThrowing" throwing="error"

pointcut="execution(public void validate())"/>

</aop:aspect>

</aop:config>

</beans>

**● Code:(App.java)**

package aop.practical4prog\_2;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class App

{

public static void main( String[] args )

{

ApplicationContext context = newClassPathXmlApplicationContext("config.xml");

Student s =(Student) context.getBean("student");

s.display();

s.validate(19);

}

}

**Output :**

Id : 87

Name : Lalita

Method was successfully executed

Error occure

**Practical 9.1**

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

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Movie.java**

package AOP.program9;

import org.springframework.jdbc.datasource.DriverManagerDataSource;

import com.mysql.jdbc.Driver;

import org.springframework.jdbc.core.JdbcTemplate;

public class movie {

private String movieName;

private String directorName;

public String getMovieName() {

return movieName;

}

public void setMovieName(String movieName) {

this.movieName = movieName;

}

public String getDirectorName() {

return directorName;

}

public void setDirectorName(String directorName) {

this.directorName = directorName;

}

public movie(String movieName, String directorName) {

super();

this.movieName = movieName;

this.directorName = directorName;

}

public movie() {

super();

// TODO Auto-generated constructor stub

}

}

**MovieDAO:**

package AOP.program9;

import org.springframework.jdbc.core.JdbcTemplate;

public class MovieDAO {

JdbcTemplate jdbcTemplate;

public void setJdbcTemplate(JdbcTemplate jdbcTemplate) {

this.jdbcTemplate = jdbcTemplate;

}

public int insertMovie(movie m) {

//insert into Movie values('RR','Rajkumar')

String query="insert into Movie

values('"+m.getMovieName()+"','"+m.getDirectorName()+"')";

return jdbcTemplate.update(query);

}

public int updateMovie(movie m) {

//update

Movie set director\_name = 'Rajkumar Rao' where name ='BB'

String query = "update Movie set

director\_name='"+m.getDirectorName()+"' where name='"+m.getMovieName()+"'";

return jdbcTemplate.update(query);

}

public int deleteMovie() {

String query = "delete from Movie where name ='BB'";

return jdbcTemplate.update(query);

}

}

**Config.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.xsd">

<bean id= "ds" class =

"org.springframework.jdbc.datasource.DriverManagerDataSource">

<property name ="driverClassName" value="com.mysql.jdbc.Driver"></property>

<property name= "url" value="jdbc:mysql://localhost:3306/theatre"></property>

<property name="username" value = "root"></property>

<property name="password" value = "qwedsa@123"></property>

</bean>

<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">

<property name= "dataSource" ref = "ds"></property>

</bean>

<bean id= "moviedao" class="AOP.program9.MovieDAO">

87-2022

Satyam Ojha

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

</bean>

</beans>

**POM.XML**

<project xmlns="http://maven.apache.org/POM/4.0.0"

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

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>AOP</groupId>

<artifactId>program9c</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>program9c</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>4.0.2.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>5.3.6</version>

</dependency>

</dependencies>

</project>

**App.java**:

package AOP.program9;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class App

{

public static void main( String[] args )

{

ApplicationContext con = new ClassPathXmlApplicationContext("config.xml");

MovieDAO mdao = (MovieDAO) con.getBean("moviedao");

movie m1 = new movie("Signham","Rohit Shetty");

movie m2 = new movie("TENET","James Gunn");

mdao.insertMovie(m1);

System.out.println("Movie Added Successfully.");

mdao.updateMovie(m2);

System.out.println("Movie updated Successfully.");

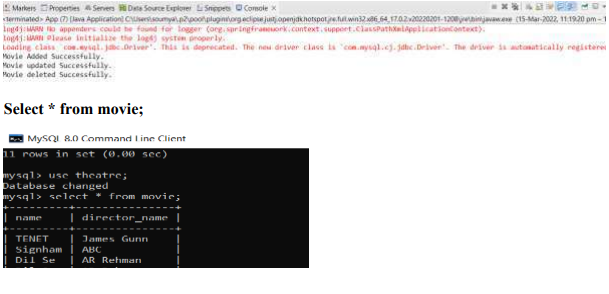
mdao.deleteMovie();

System.out.println("Movie deleted Successfully.");

}

}

**Output :**



**Practical NO : 9.2**

**Aim : Write a program to demonstrate PreparedStatement in Spring JdbcTemplate**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Movie.java:**

package AOP.program9b;

import org.springframework.jdbc.datasource.DriverManagerDataSource;

import com.mysql.jdbc.Driver;

import org.springframework.jdbc.core.JdbcTemplate;

public class movie {

private String movieName;

private String directorName;

public String getMovieName() {

return movieName;

}

public void setMovieName(String movieName) {

this.movieName = movieName;

}

public String getDirectorName() {

return directorName;

}

public void setDirectorName(String directorName) {

this.directorName = directorName;

}

public movie(String movieName, String directorName) {

super();

this.movieName = movieName;

this.directorName = directorName;

}

public movie() {

super();

// TODO Auto-generated constructor stub

}

}

**MovieDAO.java:**

package AOP.program9b;

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 MovieDAO {

JdbcTemplate jdbcTemplate;

public void setJdbcTemplate(JdbcTemplate jdbcTemplate) {

this.jdbcTemplate = jdbcTemplate;

}

public Boolean addMovie(final movie m) {

String query = "insert into Movie values(?,?)";

return jdbcTemplate.execute(query, new

PreparedStatementCallback<Boolean>() {

public Boolean doInPreparedStatement(PreparedStatement ps)

throws SQLException, DataAccessException {

// TODO Auto-generated method stub

ps.setString(1, m.getMovieName());

ps.setString(2, m.getDirectorName());

return ps.execute();

}

});

}

public Boolean updateMovie(final movie m) {

String query = "UPDATE Movie SET director\_name=? WHERE

name=?;";

return jdbcTemplate.execute(query, new

PreparedStatementCallback<Boolean>() {

public Boolean doInPreparedStatement(PreparedStatement ps)

throws SQLException, DataAccessException {

// TODO Auto-generated method stub

ps.setString(1, m.getDirectorName());

ps.setString(2, m.getMovieName());

return ps.execute();

}

});

}

public Boolean deleteMovie(final movie m) {

String query = "DELETE FROM Movie WHERE NAME=?;";

return jdbcTemplate.execute(query, new

PreparedStatementCallback<Boolean>() {

public Boolean doInPreparedStatement(PreparedStatement ps)

throws SQLException, DataAccessException {

// TODO Auto-generated method stub

ps.setString(1, "ABC");

return ps.execute();

}

});

}

}

**App.Java:**

package AOP.program9b;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class App

{

public static void main( String[] args )

{

ApplicationContext con = new ClassPathXmlApplicationContext("config.xml");

MovieDAO mdao = (MovieDAO) con.getBean("moviedao");

movie m1 = new movie("Dil Se","AR Rehman");

mdao.addMovie(m1);

System.out.println("Insert Added Successfully");

movie m2 = new movie("Signham","ABC");

mdao.updateMovie(m2);

System.out.println("Updated Successfully");

movie m3 = new movie();

mdao.deleteMovie(m3);

System.out.println("Deleted Successfully");

}

}

**Pom.xml:**

<project xmlns="http://maven.apache.org/POM/4.0.0"

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

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>AOP</groupId>

<artifactId>program9c</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>program9c</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>4.0.2.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>5.3.6</version>

</dependency>

</dependencies>

</project>

**Config.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.xsd">

<bean id= "ds" class =

"org.springframework.jdbc.datasource.DriverManagerDataSource">

<property name ="driverClassName" value="com.mysql.jdbc.Driver"></property>

<property name= "url" value="jdbc:mysql://localhost:3306/theatre"></property>

<property name="username" value = "root"></property>

<property name="password" value = "qwedsa@123"></property>

</bean>

<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">

<property name= "dataSource" ref = "ds"></property>

</bean>

<bean id= "moviedao" class="AOP.program9b.MovieDAO">

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

</bean>

</beans>

**Output :**



**Practical NO : 9.3**

**Aim : Write a program in Spring JDBC to demonstrate ResultSetExtractor Interface**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**movie.Java:**

package AOP.program9c;

import org.springframework.jdbc.datasource.DriverManagerDataSource;

import com.mysql.jdbc.Driver;

import org.springframework.jdbc.core.JdbcTemplate;

public class movie {

private String movieName;

private String directorName;

public String getMovieName() {

return movieName;

}

public void setMovieName(String movieName) {

this.movieName = movieName;

}

public String getDirectorName() {

return directorName;

}

public void setDirectorName(String directorName) {

this.directorName = directorName;

}

public movie(String movieName, String directorName) {

super();

this.movieName = movieName;

this.directorName = directorName;

}

public movie() {

super();

}

}

**MovieDAO.java**:

package AOP.program9c;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import org.springframework.dao.DataAccessException;

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.jdbc.core.ResultSetExtractor;

public class MovieDAO {

JdbcTemplate jdbcTemplate;

public void setJdbcTemplate(JdbcTemplate jdbcTemplate) {

this.jdbcTemplate = jdbcTemplate;

}

public List<movie> getMovies() {

String sql = "select \* from Movie";

return jdbcTemplate.query(sql, new

ResultSetExtractor<List<movie>>() {

public List<movie> extractData(ResultSet rs) throws

SQLException, DataAccessException {

// TODO Auto-generated method stub

List<movie> lmovie = new ArrayList<movie>();

while(rs.next()) {

movie m = new movie();

m.setMovieName(rs.getString(1));

m.setDirectorName(rs.getString(2));

lmovie.add(m);

}

return lmovie;

}

});

}

}

**pom.xml:**

<project xmlns="http://maven.apache.org/POM/4.0.0"

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

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>AOP</groupId>

<artifactId>program9c</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>program9c</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>4.0.2.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>5.3.6</version>

</dependency>

</dependencies>

</project>

**Config.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.xsd">

<bean id= "ds" class =

"org.springframework.jdbc.datasource.DriverManagerDataSource">

<property name ="driverClassName" value="com.mysql.jdbc.Driver"></property>

<property name= "url" value="jdbc:mysql://localhost:3306/theatre"></property>

<property name="username" value = "root"></property>

<property name="password" value = "qwedsa@123"></property>

</bean>

<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">

<property name= "dataSource" ref = "ds"></property>

</bean>

<bean id= "moviedao" class="AOP.program9c.MovieDAO">

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

</bean>

</beans>

**App.java:**

package AOP.program9c;

import java.util.List;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class App

{

public static void main( String[] args )

{

ApplicationContext con = new ClassPathXmlApplicationContext("config.xml");

MovieDAO mdao = (MovieDAO) con.getBean("moviedao");

List<movie> movielist = mdao.getMovies();

for(movie mov : movielist) {

System.out.print(mov.getMovieName());

System.out.println(" " +mov.getDirectorName());

}

}

}

**Output :**



**Practical NO : 9.4**

**Aim :** **Write a program to demonstrate RowMapper interface to fetch the records from the database.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**movie.java:**

package AOP.program9d;

import org.springframework.jdbc.datasource.DriverManagerDataSource;

import com.mysql.jdbc.Driver;

import org.springframework.jdbc.core.JdbcTemplate;

public class movie {

private String movieName;

private String directorName;

public String getMovieName() {

return movieName;

}

public void setMovieName(String movieName) {

this.movieName = movieName;

}

public String getDirectorName() {

return directorName;

}

public void setDirectorName(String directorName) {

this.directorName = directorName;

}

public movie(String movieName, String directorName) {

super();

this.movieName = movieName;

this.directorName = directorName;

}

public movie() {

super();

}

}

**MovieDAO:**

package AOP.program9d;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.List;

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.jdbc.core.RowMapper;

public class MovieDAO {

JdbcTemplate jdbcTemplate;

public void setJdbcTemplate(JdbcTemplate jdbcTemplate) {

this.jdbcTemplate = jdbcTemplate;

}

public List<movie> getAllMovies() {

String sql = "select \* from Movie";

List<movie> movieList = jdbcTemplate.query(sql, new

RowMapper<movie>() {

public movie mapRow(ResultSet rs, int rowNum) throws

SQLException {

movie m = new movie();

m.setMovieName(rs.getString(1));

m.setDirectorName(rs.getString(2));

return m;

}

});

return movieList; }

}

**Pom.xml:**

<project xmlns="http://maven.apache.org/POM/4.0.0"

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

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>AOP</groupId>

<artifactId>program9d</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>program9d</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>4.0.2.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>5.3.6</version>

</dependency>

</dependencies>

</project>

**Config.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.xsd">

<bean id= "ds" class =

"org.springframework.jdbc.datasource.DriverManagerDataSource">

<property name ="driverClassName" value="com.mysql.jdbc.Driver"></property>

<property name= "url" value="jdbc:mysql://localhost:3306/theatre"></property>

<property name="username" value = "root"></property>

<property name="password" value = "qwedsa@123"></property>

</bean>

<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">

<property name= "dataSource" ref = "ds"></property>

</bean>

<bean id= "moviedao" class="AOP.program9d.MovieDAO">

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

</bean>

</beans>

**App.java:**

package AOP.program9d;

import java.util.List;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class App

{

public static void main( String[] args )

{

ApplicationContext con = new ClassPathXmlApplicationContext("config.xml");

MovieDAO mdao = (MovieDAO) con.getBean("moviedao");

List<movie> movielist = mdao.getAllMovies();

for(movie mov : movielist) {

System.out.print(mov.getMovieName());

System.out.println(" " +mov.getDirectorName());

}

}

**Output:**



**Practical No : 10.1**

**Aim :** **Write a program to create a simple Spring Boot application that prints a message.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**SpringBootDemoApplication.java**

package com.example.demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.EnableAutoConfiguration;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.web.bind.annotation.RestController;

import org.springframework.web.bind.annotation.RequestMapping;

@RestController

@EnableAutoConfiguration

@SpringBootApplication

public class SpringBootDemoApplication {

public static void main(String[] args)

{

SpringApplication.run(SpringBootDemoApplication.class,args);

}

@RequestMapping("/")

public String home()

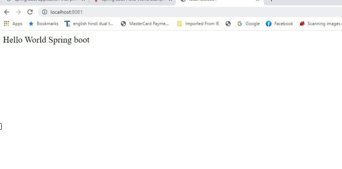
{

return "Hello World Spring boot";

}

}

**Output:**



**Practical No: 10.2**

**Aim : Write a program to demonstrate RESTful Web Services with spring boot.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Product.java:**

**package** com.example.demo;

**public class** Product {

**private** String id;

**private** String name;

**public** String getId() {

**return** id;

}

**public void** setId(String id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public void** setName(String name) {

**this**.name = name;

}

}

**ProductServiceController.java:**

package com.example.demo;

import java.util.HashMap;

import java.util.Map;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class ProductServiceController

{

private static Map<String, Product> productRepo = new

HashMap<>();

static {

Product honey = new Product();

honey.setId("1");

honey.setName("Honey");

productRepo.put(honey.getId(), honey);

Product almond = new Product();

almond.setId("2");

almond.setName("Almond");

productRepo.put(almond.getId(), almond);

}

@RequestMapping(value = "/products")

public ResponseEntity<Object> getProduct() {

return new ResponseEntity<>(productRepo.values(), HttpStatus.OK);

}

@RequestMapping(value = "/products/{id}", method = RequestMethod.PUT)

public ResponseEntity<Object> updateProduct(@PathVariable("id") String id, @RequestBody

Product product) {

productRepo.remove(id);

product.setId(id);

productRepo.put(id, product);

return new ResponseEntity<>("Product is updated successsfully", HttpStatus.OK);

}

@RequestMapping(value = "/products/{id}", method = RequestMethod.DELETE)

public ResponseEntity<Object> delete(@PathVariable("id") String id) {

productRepo.remove(id);

return new ResponseEntity<>("Product is deleted successsfully", HttpStatus.OK);

}

@RequestMapping(value = "/products", method = RequestMethod.POST)

public ResponseEntity<Object> createProduct(@RequestBody Product product) {

productRepo.put(product.getId(), product);

return new ResponseEntity<>("Product is created successfully", HttpStatus.CREATED);

}

}

**SpringBootRestfulServiceDemoApplication.java**

package com.example.demo;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringBootRestfulServiceDemoApplication {

public static void main(String[] args) {

SpringApplication.run(SpringBootRestfulServiceDemoApplication.class, args);

}

}

**Output:**

