

**LBHSS's**  
**Hiray Institute of Computer Application**

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

No	Table of Contents	
1.	<b>Practical on Java Generics</b>	
	1	<a href="#">Write a Java Program to demonstrate a Generic Class.</a>
	2	<a href="#">Write a Java Program to demonstrate Generic Methods.</a>
	3	<a href="#">Write a Java Program to demonstrate Generic Method –part 2</a>
	4	<a href="#">Write Program in java to understand the use of unbound wildcards</a>
	5	<a href="#">Write Program in java to understand the use of upper bound wildcards</a>
2	<b>Practical on List Interface</b>	
	1	<a href="#">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.</a>
	2	<a href="#">Write a Java program to create List containing list of items and use ListIterator interface to print items present in the list. Also print the list in reverse/ backward`direction.</a>
3	<b>Practical on set Interface</b>	
	1	<a href="#">Write a Java program to create a Set containing list of items of type String and printthe items in the list using Iterator interface. Also print the list in reverse/ backward direction.</a>
	2	<a href="#">Write a Java program using Set interface containing list of items and perform thefollowing operations:</a> a. <a href="#">Add items in the set.</a> b. <a href="#">Insert items of one set in to other set.</a> c. <a href="#">Remove items from the set</a> d. <a href="#">Search the specified item in the set</a>
4	<b>Practical on Map Interface</b>	
	1	<a href="#">Write a Java program using Map interface containing list of items having keys andassociated values and perform the following operations:</a> a. <a href="#">Add items in the map.</a> b. <a href="#">Remove items from the map</a> c. <a href="#">Search specific key from the map</a> d. <a href="#">Get value of the specified key</a> e. <a href="#">Insert map elements of one map in to other map.</a> f. <a href="#">Print all keys and values of the map.</a>
5	<b>Practical on Lambda Expression</b>	
	1	<a href="#">Write a Java program using Lambda Expression to print "Hello World".</a>
	2	<a href="#">Write a Java program using Lambda Expression with single parameters.</a>
	3	<a href="#">Write a Java program using Lambda Expression with multiple parameters to add twonumbers.</a>
	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	<a href="#">Write a Java program using Lambda Expression to concatenate two strings.</a>

6	<b>Practical based on web application development using JSP</b>	
	1	<a href="#">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.</a>
	2	<a href="#">Write a JSP page to display the Registration form (Make your own assumptions)</a>
	3	<a href="#">Write a JSP program to add, delete and display the records from StudentMaster (RollNo, Name, Semester, Course) table</a>
	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: <ul style="list-style-type: none"> <li>a. 1 to 7 year at 5.35%</li> <li>b. 8 to 15 year at 5.5%</li> <li>c. 16 to 30 year at 5.75%</li> </ul>
	5	<a href="#">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</a>
	6	<a href="#">Write a JSP program to add, delete and display the records from StudentMaster (RollNo, Name, Semester, Course) table.</a>
	7	Write a JSP program that demonstrates the use of JSP declaration, scriptlet, directives, expression, header and footer.
7	<b>Practical on Spring Framework</b>	
	1	<a href="#">Write a program to print "Hello World" using spring framework.</a>
	2	<a href="#">Write a program to demonstrate dependency injection via setter method.</a>
	3	<a href="#">Write a program to demonstrate dependency injection via setter method.</a>
	4	<a href="#">Write a program to demonstrate dependency injection via Constructor.</a>
8	<b>Practical based on Aspect Oriented Programming</b>	
	0	<a href="#">Write a program to demonstrate Spring AOP – All</a>
	1	<a href="#">Write a program to demonstrate Spring AOP – before advice.</a>
	2	<a href="#">Write a program to demonstrate Spring AOP – after advice</a>
	3	Write a program to demonstrate Spring AOP – around advice.
	4	Write a program to demonstrate Spring AOP – after returning advice. [Refer 0]
	5	Write a program to demonstrate Spring AOP – after throwing advice. [Refer 0]
	6	Write a program to demonstrate Spring AOP – pointcuts. [Refer 0]
9	<b>Practical based on SPRING JDBC</b>	
	1	<a href="#">Write a program to insert, update and delete records from the given table.</a>
	2	Write a program to demonstrate PreparedStatement in Spring JdbcTemplate [Refer 1]
	3	<a href="#">Write a program in Spring JDBC to demonstrate ResultSetExtractor Interface</a>
	4	<a href="#">Write a program to demonstrate RowMapper interface to fetch the records from the database.</a>
10	<b>Practical based on SPRING boot and RESTFUL Web Services</b>	

	1	<a href="#">Write a program to create a simple Spring Boot application that prints a message.</a>
	2	<a href="#">Write a program to demonstrate RESTful Web Services with spring boot.</a>

## Practical 1.1

---



```
/* Understand the Generic class
 * Write a Java Program to demonstrate a Generic Class
 * Author: Bhanudas Satam
 * rollno:
 * div:
 * */
```

### Solution

---

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

    }
}
```

### Output

---

```
Variable value=10
Variable value=Sukhiram
```

## Practical 1.2



```
/*
 * Practical b1
 * Write a program in java that will use Generic methods
 * There are following kind of Generic Types
 *   T Type
 *   E Element
 *   k Key
 *   v Value
 *   N Number
 */
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

```
1,2,3,4,5,6,
1.45,2.35,4.63,8.56,
H,E,L,L,\0,
```

## Practical 1.2a



```
/*
 * Write a program in java that will use Generic methods
 * There are following kind of Generic Types
 *   T   Type
 *   E   Element
 *   k   Key
 *   v   Value
 *   N   Number
 *   understand the use of k,v types
 *   S,V
 */
class Pair<K,V>
{
    private K key;
    private V value;

    public Pair(K key, V value) {
        this.key = key;
        this.value = value;
    }
    public K getKey() {
        return key;
    }
    public void setKey(K key) {
        this.key = key;
    }
    public V getValue() {
        return value;
    }
    public void setValue(V value) {
        this.value = value;
    }
}

public class GenericMethod2 {

    public static <K,V> boolean compare(Pair<K,V>p1,Pair<K,V>p2) {
        return p1.getKey().equals(p2.getKey()) &&
p1.getValue().equals(p2.getValue());
    };
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Pair<Integer,String>p1=new Pair<>(1,"aaa");
        Pair<Integer,String>p2=new Pair<>(2,"bbb");
        boolean result=GenericMethod2.compare(p1, p2);
        System.out.println(result);
    }
}
```

### Output

false

## Practical 1.3



Write a program to demonstrate unbound Wildcard in List Interface

```
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

/*
 * Write a program in java to understand the use
 * of unbound wildcards ?
 */
public class unboundWildcardExample {
    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) {
        // TODO Auto-generated method stub
        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("Suresh");
        mynameList.add("Ramesh");
        mynameList.add("Jayesh");
        mynameList.add("Mahesh");
        printArray(mynameList);
    }
}
```

### Output

```
1
2
3
-----
Suresh
Ramesh
Jayesh
Mahesh
-----
```



## Practical 1.3a



Write a program to demonstrate upper Wildcard in List Interface

```
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

/*
 * Write a program in java to understand the use
 * of upperbound wildcards ? extends
 */
public class upperboundWildcardExample {
    public static void sumofElements(List<? extends Number> numberlist)
    {
        double sum=0.0;
        for (Number n : numberlist) sum += n.doubleValue();
        System.out.println("Sum of all elements is="+sum);
    }
    public static void main(String []args) {
        ArrayList <Integer>intlist=new ArrayList<Integer>();
        intlist.add(1);
        intlist.add(2);
        intlist.add(3);
        sumofElements(intlist);
        ArrayList <Double>dblist=new ArrayList<Double>();
        dblist.add(24.56);
        dblist.add(21.24);
        dblist.add(124.16);
        dblist.add(121.14);
        sumofElements(dblist);
    }
}
```

### Output

Sum of all elements is=6.0

Sum of all elements is=291.09999999999997

## Practical 2.1

---

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

```
import java.util.ArrayList;
import java.util.List;

/* Practical 2-1
 * Author: Bhanudas Satam
 *
 * 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.
 */
public class ListInterfaceExample1 {

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

---

Prakash  
Suresh  
Ramesh  
Jayesh

## Practical 2.2

---

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

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.Iterator;
import java.util.List;

/*
 * Write a Java program to create List containing list of items
 * and use List Iterator interface to print items present in the
 * list. Also print the list in reverse/ backward direction.
 */
public class ListInterfaceRevFor {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        List<Integer> numberlist=new ArrayList<Integer>();
        //Generating list of 10 numbers
        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

---

1  
2

3  
4  
5

6  
7  
8  
9  
10  
Reverse order  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1

## Practical 3.1

---

### Problem Statement

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

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.ArrayList;
import java.util.Collection;
import java.util.Collections;
import static java.util.Collections.list;
import java.util.Comparator;
import java.util.HashSet;
import java.util.Iterator;
import java.util.List;
import java.util.ListIterator;
import java.util.NavigableSet;
import java.util.Scanner;
import java.util.TreeSet;

/**
 *
 * @author BHANUDAS SATAM
 */
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::getNa
            me,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
}
}

```

## Practical 3.2

---

### Problem Statement

---

1. Write a Java program using Set interface containing list of items and perform the following operations:
  1. Add items in the set.
  2. Insert items of one set in to other set.
  3. Remove items from the set
  4. Search the specified item in the set

### 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.HashSet;
import java.util.Iterator;
import java.util.Scanner;
import java.util.function.Predicate;

/**
 *
 * @author BHANUDAS SATAM
 */
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.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

---

Menu

- 1.Adding student
- 2.Removing student
- 3.Search student
- 4.List all students
- 5.Exit

Enter your choice(1..4)1

Enter rollno 1

Enter name AAA

Enter percentage 99.98

Do you wish to continue(y/n)y

Menu

- 1.Adding student
- 2.Removing student
- 3.Search student
- 4.List all students
- 5.Exit

Enter your choice(1..4)1

Enter rollno 2  
Enter name BBB  
Enter percentage 89.45  
Do you wish to continue(y/n)y  
Menu  
1.Adding student  
2.Removing student  
3.Search student  
4.List all students  
5.Exit  
Enter your choice(1..4)1  
Enter rollno 3  
Enter name CCC  
Enter percentage55.24  
Do you wish to continue(y/n)y  
Menu  
1.Adding student  
2.Removing student  
3.Search student  
4.List all students  
5.Exit  
Enter your choice(1..4)3  
Enter the name of student  
CCC  
Record found  
3 CCC 55.24  
Do you wish to continue(y/n)y  
Menu  
1.Adding student  
2.Removing student  
3.Search student  
4.List all students  
5.Exit  
Enter your choice(1..4)4  
Rollno Name Percentage  
3 CCC 55.24  
2 BBB 89.45  
1 AAA 99.98

Do you wish to continue(y/n)y

Menu

1.Adding student

2.Removing student

3.Search student

4.List all students

5.Exit

Enter your choice(1..4)2

Enter roll no to remove element2

Rollno=2

Element is successfully removed

Do you wish to continue(y/n)y

Menu

1.Adding student

2.Removing student

3.Search student

4.List all students

5.Exit

Enter your choice(1..4)4

Rollno Name Percentage

3 CCC 55.24

1 AAA 99.98

Do you wish to continue(y/n)n



## Practical 4

---

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

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

Book.java

---

```
package com.hiraymca;

/**
 *
 * @author BHANUDAS SATAM
 */
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.HashMap;
import java.util.Map;
import java.util.Scanner;

/**
 *
 * @author BHANUDAS SATAM
 */
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);
    }
}

```

```

//Traversing the map

for(Map.Entry<Integer, Book> entry:map.entrySet()){
    int key=entry.getKey();
    b=entry.getValue();
    System.out.println(b.getId()+" "+b.getName()+" "+b.getAuthor());
}

//Removing element from map
map.remove(2);

//Traversing the map after removing
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());
}

//searching for specific entry
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

---

run:

```

101 Let us C Yashwant Kanetkar
102 Data communication & Networking Forouzan
103 Operating System Achuyut Godbole

```

Traversing map after removing 2nd element

101 Let us C Yashwant Kanetkar  
103 Operating System Achuyut Godbole  
Enter the key 101  
101 Let us C Yashwant Kanetkar

## Practical 5.1

---

```
/* @author BHANUDAS SATAM
 * Write a Java program using Lambda Expression with single parameters.
 */
```

```
interface HelloWorld {
    void sayHello();
}

public class LambdaExpressionExample {
    public static void main(String args[]){
        HelloWorld helloworld=()->{System.out.println("Hello World");};
        helloworld.sayHello();
    }
}
```

Output

---

Hello World

## Practical 5.2

---

\* Write a Java program using Lambda Expression with single parameters

```
interface findCube
{
    int cube(int num);
}
public class Practical52 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        findCube cubeobj=(int x)->{return x*x*x;};
        int result=cubeobj.cube(2);
        System.out.println("Cube of Enter number is="+result);
    }
}
```

Output

---

Cube of Enter number is=8

## Practical 5.3

---

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

\*/

```
interface Calculator
```

```
{
```

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

```
}
```

```
public class Practical53 {
```

```
    /**
```

```
     * @param args the command line arguments
```

```
     */
```

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

```
        // TODO code application logic here
```

```
        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 is =" +result);
```

```
    }
```

```
}
```

Output

---

Addition of two number is =30

## Practical 5.6

---

```
interface Constr{
    String addstr(String fname,String lname);
}

public class Practical56 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        String fname="Bhanudas";
        String lname="Satam";
        Constr ctr=(String f,String l)->{return f+" "+l;};
        String fullstring=ctr.addstr(fname,lname);
        System.out.println("My Full name is "+fullstring);
    }

}
```

Output

---

My Full name is Bhanudas Satam



# Practical 6.1

---

## Index.jsp

---

<%--

Document : index

Created on : Mar 22, 2021, 12:25:31 PM

Author : BHANUDAS SATAM

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

--%>

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

<!DOCTYPE html>

<html>

<head>

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

<title>Practical 6-1</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

---

Telephone-Entry-Form	
Name	<input type="text"/>
Telephone Number	<input type="text"/>
<input type="button" value="Add Entry"/>	<input type="button" value="Reset"/>

## Process.jsp

---

```

<%--
    Document : Process
    Created on : Mar 22, 2021, 11:07:53 AM
    Author    : BHANUDAS SATAM
--%>

<%@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</title>
  </head>
  <body>
    <%
      String url="jdbc:mysql://localhost:3306/studentdb?zeroDateTimeBehavior=convertToNull";
      String user="root";
      String password="";
      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(url, user, password);
      try
      {

        //checking the record already exist in database or not
        st=conn.createStatement();
        rs=st.executeQuery("select * from teldir");
        while(rs.next())
        {
          if(rs.getString(2).equals(uname)&& rs.getString(3).equals(tel))
          {
            <%>
            <jsp:forward page="index.jsp">
              <jsp:param name="message" value="User is already exists, Give New Entry !!!" />
            </jsp:forward>
            <%
              }
            }
          //inserting record in database
          String sql="insert into teldir(name,telephone) values(?,?)";
          ps=conn.prepareStatement(sql);
          ps.setString(1,uname);
          ps.setString(2,tel);
          ps.executeUpdate();
        }
      }
    %>
  </body>
</html>

```

```

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='index.jsp'>here</a>"+ "to Enter another record");
out.println("</h3>");

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

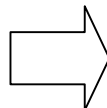
```

## Output

---

If user is already exist

Telephone-Entry-Form	
Name	<input type="text" value="Bhanudas Kashinath Satam"/>
Telephone Number	<input type="text" value="9869281034"/>
<input type="button" value="Add Entry"/>	<input type="button" value="Reset"/>



User is already exists, Give New Entry !!!

Telephone-Entry-Form	
Name	<input type="text"/>
Telephone Number	<input type="text"/>
<input type="button" value="Add Entry"/>	<input type="button" value="Reset"/>

New Entry

Telephone-Entry-Form	
Name	<input type="text" value="Suresh Sawant"/>
Telephone Number	<input type="text" value="9869349521"/>
<input type="button" value="Add Entry"/>	<input type="button" value="Reset"/>

---

**Record inserted successfully**  
 Click [here](#) to Enter another record

## Practical 6.2

---

Index.jsp

---

```
<%--
    Document : index
    Created on : Mar 19, 2021, 12:54:11 PM
    Author    : BHANUDAS SATAM
    Write a JSP page to display the Registration form
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Registration Form </title>
        <link rel="stylesheet" href="https://code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
        <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
        <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
        <script>
            $( function() {
                $.datepicker.setDefaults({
                    onClose:function(date, inst){
                        $("#selectedDtaeVal").html(date);
                    }
                });
                $( "#datepicker" ).datepicker();});
        </script>
    </head>
    <body>
        <form action="process.jsp">
            <table border="1" cellpadding="10px" align="center">
                <thead>
                    <tr>
                        <th colspan="2" style="background-color: yellow" padding="10px">Registration Form</th>
                    </tr>
                </thead>
                <tbody>
                    <tr>
                        <td>Full Name</td>
                        <td><input type="text" name="txtName" value="" /></td>
                    </tr>
                </tbody>
            </table>
        </form>
    </body>
</html>
```

```

<tr>
  <td>Email</td>
  <td><input type="text" name="txtEmail" value="" /></td>
</tr>
<tr>
  <td>Date of Birth</td>
  <td><input type="text" name="txtbdate" id="datepicker"></td>
</tr>
<tr>
  <td>Educational Qualification</td>
  <td><select name="opt-qualify">
    <option>HSC</option>
    <option>Diploma</option>
    <option>Graduate</option>
    <option>Post-Graduate</option>
  </select></td>
</tr>
<tr>
  <td>Languages Known</td>
  <td><input type="checkbox" name="lk" value="English" />English
    <input type="checkbox" name="lk" value="Marathi" />Marathi
    <input type="checkbox" name="lk" value="Hindi" />Hindi
  </td>
</tr>
<tr>
  <td valign="top">Write about yourself</td>
  <td><textarea name="about" rows="8" cols="30" >
    </textarea></td>
</tr>
<tr>
  <td>Gender </td>
  <td><input type="radio" name="gen" value="Male" />Male
    <input type="radio" name="gen" value="Female" />Female
    <input type="radio" name="gen" value="Tgen" />Tgen</td>
</tr>
<tr>
  <td><input type="submit" value="Submit" /></td>
  <td><input type="reset" value="Reset" /></td>
</tr>
</tbody>
</table>

```

</form>

</body>

</html>

Output

---

Registration Form	
Full Name	<input type="text" value="Bhanudas Kashinath Satam"/>
Email	<input type="text" value="bhanudas.satam@gmail.com"/>
Date of Birth	<input type="text" value="02/19/1975"/>
Educational Qualification	<input type="text" value="Post-Graduate"/>
Languages Known	<input checked="" type="checkbox"/> English <input checked="" type="checkbox"/> Marathi <input checked="" type="checkbox"/> Hindi
Write about yourself	<div>I am good in java programming. I will rate myself as 4 in the range of 1 to 5</div>
Gender	<input checked="" type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Tgen
<input type="button" value="Submit"/>	<input type="button" value="Reset"/>

process.jsp

---

```
<%--
    Document : process
    Created on : Mar 19, 2021, 2:16:48 PM
    Author    : BHANUDAS SATAM
--%>

<%@page import="java.util.Arrays"%>
<%@page import="java.util.List"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Practical6-2</title>
    </head>
    <body>
        <h1>You are successfully Registered </h1>
        <h4>Your details are as follows </h4>
        <% String name,email,dob,equalify,lang,yourself,gen;
            name=request.getParameter("txtName");
            email=request.getParameter("txtEmail");
            dob=request.getParameter("txtbdate");
            equalify=request.getParameter("opt-qualify");
            lang=request.getParameter("English");
            yourself=request.getParameter("about");
            gen=request.getParameter("gen");

            String [] lknown=request.getParameterValues("lk");
            List list=Arrays.asList(lknown);
            request.setAttribute("lknown",list);
            List<String> lk = (List<String>)request.getAttribute("lknown");
            out.println("<table border='1' cellpadding='15px'>");

            out.print("<tr>");
            out.print("<td>");
            out.print("Name");
            out.print("</td>");
            out.print("<td>");
            out.print(name);
```



```
out.print("</td>");  
out.print("</tr>");
```

```
out.print("<tr>");  
out.print("<td>");  
out.print("Email");  
out.print("</td>");  
out.print("<td>");  
out.print(email);  
out.print("</td>");  
out.print("</tr>");
```

```
out.print("<tr>");  
out.print("<td>");  
out.print("Date of Birth");  
out.print("</td>");  
out.print("<td>");  
out.print(dob);  
out.print("</td>");  
out.print("</tr>");
```

```
out.println("<tr>");  
out.println("<td>");  
out.println("Educational Qualification");  
out.println("</td>");  
out.println("<td>");  
out.println(equalify);  
out.println("</td>");  
out.println("</tr>");
```

```
out.println("<tr>");  
out.println("<td>");  
out.println("Gender");  
out.println("</td>");  
out.println("<td>");  
out.println(gen);  
out.println("</td>");  
out.println("</tr>");
```

```
out.println("<tr>");  
out.println("<td>");  
out.println("Languages 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("About yourself");
        out.println("</td>");
        out.println("<td>");
        out.println(yourself);
        out.println("</td>");
        out.println("</tr>");
        out.println("</table>");
    %>
</body>
</html>

```

Output

---

## You are successfully Registered

Your details are as follows

Name	Bhanudas Kashinath Satam
Email	bhanudas.satam@gmail.com
Date of Birth	02/19/1975
Educational Qualification	Post-Graduate
Gender	Male
Languages known	English Marathi Hindi
About yourself	I am good in java programming. I will rate myself as 4 in the range of 1 to 5

## Practical 6.3

---

### add.jsp

---

```
<%--
    Document : index
    Created on : Mar 18, 2021, 8:49:16 AM
    Author    : BHANUDAS SATAM
--%>

<%@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;
            }
            #tablehead
            {
                background-color: yellow;
                padding:10px;
            }
        </style>
    </head>
    <body>

        <%@include file="header.jsp" %>
        <div id="content">
            <h2 align="center">Adding Record </h2>
            <table align="center">
                <tr align="center" style="color:blue">
                    <td>${param.message}</td>
                </tr>
            </table>
            <form action="addrecord.jsp">
                <table border="1" align="center" cellpadding="10px">
                    <thead>
                        <tr id='tablehead'>
                            <th colspan="2">Student Master Form</th>
```

```

        </tr>
    </thead>
    <tbody>

        <tr>
            <td>Name</td>
            <td><input type="text" name="txtname" value="" /></td>
        </tr>
        <tr>
            <td>Semester</td>
            <td><select name="optsem">
                <option>Sem I</option>
                <option>Sem II</option>
                <option>Sem III</option>
                <option>Sem IV</option>
            </select></td>
        </tr>
        <tr>
            <td>Course</td>
            <td><select name="optcourse">
                <option>MCA</option>
                <option>M.Sc[Comp.Sc]</option>
                <option>M.Sc[I.T.]</option>
            </select></td>
        </tr>
        <tr>
            <td><input type="submit" value="Add Record" /></td>
            <td><input type="reset" value="Reset" /></td>
        </tr>
    </tbody>
</table>

</form>
<%@include file="footer.jsp" %>
</body>
</html>

```

Addrecord.jsp

---

```
<%--
```

```
    Document : addrecord
```

```
    Created on : Mar 18, 2021, 8:57:40 AM
```

Author : BHANUDAS SATAM

--%>

```
<%@page import="java.sql.ResultSetMetaData"%>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.DriverManager"%>
<%@page import="java.sql.PreparedStatement"%>
<%@page import="java.sql.Connection"%>
<%@page import="java.sql.ResultSet"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Add Record process</title>
  </head>
  <body>
    <%
      String sname=request.getParameter("txtname");
      String sem=request.getParameter("optsem");
      String course=request.getParameter("optcourse");
      Connection conn;
      PreparedStatement ps;
      String url="jdbc:mysql://localhost:3306/studentdb";
      String user="root";
      String password="";
      //step 1
      try
      {
        Class.forName("com.mysql.jdbc.Driver");
        conn=DriverManager.getConnection(url, user, password);
        String sql="insert into student(sname,semester,course)value(?,?,?)";

        ps=conn.prepareStatement(sql);
        ps.setString(1, sname);
        ps.setString(2, sem);
        ps.setString(3, course);
        ps.executeUpdate();
        conn.close();

      }catch(ClassNotFoundException e)
      {
```

```

        e.printStackTrace();
    }catch(Exception e)
    {
        e.printStackTrace();
    }
}%>
<jsp:forward page="add.jsp">
    <jsp:param name="message" value="record inserted successfully" />
</jsp:forward>
</body>
</html>

```

## Output

[Home](#)
[Add Record](#)
[Display Record](#)
[Delete Record](#)

### Adding Record

Student Master Form	
Name	<input type="text"/>
Semester	Sem I ▾
Course	MCA ▾
<input type="button" value="Add Record"/>	<input type="button" value="Reset"/>

© All rights reserved for Hiray Institute of computer technology

[Home](#)
[Add Record](#)
[Display Record](#)
[Delete Record](#)

### Adding Record

record inserted successfully

Student Master Form	
Name	<input type="text"/>
Semester	Sem I ▾
Course	MCA ▾
<input type="button" value="Add Record"/>	<input type="button" value="Reset"/>

© All rights reserved for Hiray Institute of computer technology

## display.jsp

---

```
<%--
    Document : index
    Created on : Mar 18, 2021, 8:49:16 AM
    Author    : BHANUDAS SATAM
--%>

<%@page import="java.sql.ResultSetMetaData"%>
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.SQLException"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.PreparedStatement"%>
<%@page import="java.sql.DriverManager"%>
<%@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>Index page</title>
    <style>
        #content
        {
            margin-top:10px;
        }
        .tablehead
        {
            background-color: yellow;
            padding:10px;
        }
    </style>
</head>
<body>
    <%
        String rollno=request.getParameter("optrollno");
        Connection conn;
        ResultSetMetaData rsmd;
        ResultSet rs1=null,rs2=null;
        Statement st;
        PreparedStatement ps;
```

```

String rn=null,nam=null,sam=null,co=null;
try
{
conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/studentdb","root","");
st=conn.createStatement();
rs1=st.executeQuery("select rollno from student");
ps=conn.prepareStatement("select * from student where rollno=?");
ps.setInt(1, Integer.parseInt(rollno));
rs2=ps.executeQuery();
rs2.next();
rn=rs2.getString(1);
nam=rs2.getString(2);
sam=rs2.getString(3);
co=rs2.getString(4);

}catch(NumberFormatException e)
{
    e.printStackTrace();
}
catch(SQLException e)
{
    e.printStackTrace();
}
}%>
<%@include file="header.jsp" %>
<div id="content">
    <h4 align="center">Selecting Record </h4>
    <form action="display.jsp" name="myform">
        <table border="1" align="center" cellpadding="10px">
            <thead>
                <tr class='tablehead'>
                    <th colspan="2">Display Record</th>

                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>Choose Rollno</td>
                    <td><select name="optrollno" onchange="document.myform.submit()">
                        <%
                            rsmd=rs1.getMetaData();
                            int tcol=rsmd.getColumnCount();

```



```

        out.println("<option>");
        out.println("Select Rollno");
        out.println("</option>");
        while(rs1.next())
        {
            out.println("<option>");
            out.println(rs1.getString(1));
            out.println("</option>");
        }
    %>
</select></td>
</tr>

</tbody>
</table>
<br>

<div id="output">
    <table border="1" align="center" cellpadding="10px">
        <thead>
            <tr>
                <th colspan="4" align="center" class="tablehead">Record </th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <th>Roll No</th>
                <th>Name</th>
                <th>Semester</th>
                <th>Course</th>
            </tr>
            <tr>
                <td><%= rn%></td>
                <td><%= nam%></td>
                <td><%= sam%></td>
                <td><%= co%></td>
            </tr>
        </tbody>
    </table>

</div>
</form>

```

```

    <%@include file="footer.jsp" %>
</body>
</html>

```

## Output

[Home](#)
[Add Record](#)
[Display Record](#)
[Delete Record](#)

Selecting Record

Display Record

Record			
Roll No	Name	Semester	Course
null	null	null	null

© All rights reserved for Hiray Institute of computer technology

[Home](#)
[Add Record](#)
[Display Record](#)
[Delete Record](#)

Selecting Record

Display Record

Record			
Roll No	Name	Semester	Course
19	Bhanudas Kashinath Satam	Sem I	MCA

© All rights reserved for Hiray Institute of computer technology

## Delete.jsp

```

<%--
Document : index
Created on : Mar 18, 2021, 8:49:16 AM
Author   : BHANUDAS SATAM
--%>

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

```

```

<%@page import="java.sql.Statement"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Connection"%>
<%@page import="java.sql.ResultSetMetaData"%>
<%@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;
        }
        .tablehead
        {
            background-color: yellow;
            padding:10px;
        }
    </style>
</head>
<body>
    <%
        String rollNo=request.getParameter("optrollno");
        Connection conn;
        ResultSetMetaData rsmd;
        ResultSet rs=null;
        Statement st;

        try
        {
            conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/studentdb","root","");
            st=conn.createStatement();
            rs=st.executeQuery("select rollno from student");
        }catch(NumberFormatException e)
        {
            e.printStackTrace();
        }
        catch(SQLException e)
        {
            e.printStackTrace();
    %>

```

```

    }
    %>
<%@include file="header.jsp" %>
<div id="content">
    <h4 align="center">Deleting Record </h4>
    <form action="deleterecord.jsp" name="myform">
        <table border="1" align="center" cellpadding="10px">
            <thead>
                <tr class='tablehead'>
                    <th colspan="2">Delete Record</th>

            </tr>
            </thead>
            <tbody>
                <tr>
                    <td>Choose Rollno</td>
                    <td><select name="optrollno" onchange="document.myform.submit();">
                        <%
                            rsmd=rs.getMetaData();
                            int tcol=rsmd.getColumnCount();
                            out.println("<option>");
                            out.println("Select Rollno");
                            out.println("</option>");
                            while(rs.next())
                            {
                                out.println("<option>");
                                out.println(rs.getString(1));
                                out.println("</option>");
                            }
                        %>
                    </select></td>

                </tr>

            </tbody>
        </table>
        <table align="center">
            <tr>
                <td colspan="2">${param.message}</td>
            </tr>
        </table>
    <%@include file="footer.jsp" %>
</body>

```

</html>

## Deleterecord.jsp

---

<%--

Document : deleterecord

Created on : Mar 18, 2021, 9:19:27 AM

Author : BHANUDAS SATAM

--%>

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

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

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

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

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

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

<%@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 rollno=request.getParameter("optrollno");

Connection conn;

PreparedStatement ps;

try

{

conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/studentdb","root","");

ps=conn.prepareStatement("delete from student where rollno = ?");

ps.setInt(1, Integer.parseInt(rollno));

ps.executeUpdate();

conn.commit();

conn.close();

}catch(NumberFormatException e)

{

e.printStackTrace();

}

catch(SQLException e)

```
        {
            e.printStackTrace();
        }

    %>
    <jsp:forward page="delete.jsp">
        <jsp:param name="message" value="Records Deleted successfully"/>
    </jsp:forward>
</body>
</html>
```

## Output

---

Home Add Record Display Record Delete Record

Deleting Record

Delete Record

Choose Rollno

All rights reserved for Hiray Institute of computer technology

## Practical 6.5

---

Index.jsp

---

<%--

Document : index

Created on : Mar 23, 2021, 7:31:15 AM

Author : BHANUDAS SATAM

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

--%>

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

<!DOCTYPE html>

<html>

<head>

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

<title>Practical 6-5</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

---

```

<%--
    Document : Process
    Created on : Mar 23, 2021, 7:36:34 AM
    Author    : BHANUDAS SATAM
--%>

<%@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>
```

## Application form for Change of Study Center

Registration id	<input type="text" value="1001"/>
Name	<input type="text" value="Bhanudas Kashinath Satam"/>
Old Study center address	<input type="text" value="NMITD Dadar"/>
New Study center address	<input type="text" value="Hiray College, Bandra"/>
<input type="button" value="Change It"/>	<input type="button" value="Reset"/>



## Address is Successfully change

Registration Id	1001
Name	Bhanudas Kashinath Satam
Old address	NMITD Dadar
New address	Hiray College, Bandra

## Practical 6.6

---

index.jsp

---

```
<%--
    Document : index
    Created on : Mar 18, 2021, 8:10:33 AM
    Author   : BHANUDAS SATAM
    Write a JSP program that demonstrates the use of JSP declaration,
    scriptlet, directives, expression, header and footer.
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Practical 6-6</title>
        <style>
            #content
            {
                margin-top:10px;
            }
        </style>
    </head>
    <body>

        <!--use of Include Directives as header -->
        <%@include file="header.jsp" %>
        <div id="content">
            <!--JSP Declaration -->
            <%! int  number1=10,number2=20;%>
            Value of number is= <%= number1%> <br>
            <!-- use of JSP scriptlet -->
            Today's Date is <% out.println(new java.util.Date()); %><br>
            <!-- use of expression -->
            Number1=<%=number1%><br>
            Number2=<%=number2%><br>
            Addition of two numbers are <%=number1+number2%><br>
        </div>
        <!--use of Include Directives as footer-->
        <%@include file="footer.jsp" %>

    </body>
```

</html>

header.jsp

---

<%--

Document : header

Created on : Mar 18, 2021, 8:24:06 AM

Author : BHANUDAS SATAM

--%>

<%@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>

<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

---

```
<%--
    Document : footer
    Created on : Mar 18, 2021, 8:16:07 AM
    Author   : BHANUDAS SATAM
--%>

<%@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>
        <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

---

[Home](#) [Service](#) [Contact us](#) [About us](#)

Value of number is- 10  
Today's Date is Thu Mar 18 08:39:53 IST 2021  
Number1=10  
Number2=20  
Addition of two numbers are 30

© All rights reserved for Hiray Institute of computer technology

## Practical 7.1

---

1 Write a program to print "Hello World" using spring

HelloWorld.java

---

```
package com.hiraymca;

public class HelloWorld {
    private String message;

    public void getMessage() {
        System.out.println("Your message="+message);
    }

    public void setMessage(String message) {
        this.message = message;
    }
}
```

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");
        HelloWorld obj=(HelloWorld)context.getBean("helloworld");
        obj.setMessage("Hello World");
        obj.getMessage();
    }
}
```

## 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"
       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>
```

## Output

---

Your message=Hello World



## Practical 7.2

---

Write a program to demonstrate dependency injection via setter method.

---

### Employee.java

---

```
package com.hiraymca;

public class Employee {
    private String ename;
    private int eage;
    private int esal;
    public String getName() {
        return ename;
    }
    public void setName(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();

    }

}

```

#### 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"
        xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
    <bean id="employee" class="com.hiraymca.Employee">
        <property name="ename" value="AAA"></property>
        <property name="eage" value="24"> </property>
        <property name="esal" value="75000"></property>
    </bean>
</beans>

```

#### Output

---

```

Name=AAA
Age=24
Salary=75000

```

## Practical 7.2b

---

Write a program to demonstrate dependency injection via setter method using beanfactory container

---

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=22;
        this.esal=65000;
    }

    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);
    }
}

```

#### 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"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
<bean id="emp" class="com.hiraymca.Employee" > </bean>

</beans>

```

#### MainApp.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();
    }
}

```

#### Output

---

```

Name=XYZ
Age=22
Salary=65000

```

## Practical 7.3

---

Write a program to demonstrate dependency injection via Constructor.

---

### 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="Dukhiram";
        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);
    }
}

```

#### MainApp.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();
    }
}

```

#### 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"
    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

---

```

Name=sukhiram
Age=65
Salary=15000

```

## Practical 8.0

---

### XML Schema Based AOP with Spring

To use the AOP namespace tags described in this section, you need to import the springAOP schema as described

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

    <!-- bean definition & AOP specific configuration -->

</beans>
```

You will also need the following AspectJ libraries on the CLASSPATH of your application. These libraries are available in the 'lib' directory of an AspectJ installation, otherwise you can download them from the internet.

- aspectjrt.jar
- aspectjweaver.jar
- aspectj.jar
- aopalliance.jar

Logging.java

---

```
package com.hiraymca;

public class Logging {
    /**
     * This is the method which I would like to execute
     * before a selected method execution.
     */
    public void beforeAdvice(){
        System.out.println("Going to setup student profile.");
    }

    /**
     * This is the method which I would like to execute
     * after a selected method execution.
     */
    public void afterAdvice(){
        System.out.println("Student profile has been setup.");
    }

    /**
     * This is the method which I would like to execute
     * when any method returns.
     */
    public void afterReturningAdvice(Object retVal) {
        System.out.println("Returning:" + retVal.toString() );
    }
}
```

```

/**
 * This is the method which I would like to execute
 * if there is an exception raised.
 */
public void AfterThrowingAdvice(IllegalArgumentException ex){
    System.out.println("There has been an exception: " + ex.toString());
}
}

```

## Student.java

```

package com.hiraymca;

public class Student {
    private Integer age;
    private String name;

    public void setAge(Integer age) {
        this.age = age;
    }
    public Integer getAge() {
        System.out.println("Age : " + age );
        return age;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getName() {
        System.out.println("Name : " + name );
        return name;
    }
    public void printThrowException(){
        System.out.println("Exception raised");
        throw new IllegalArgumentException();
    }
}

```

## 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) {
        ApplicationContext context = new
        ClassPathXmlApplicationContext("Beans.xml");

        Student student = (Student) context.getBean("student");
        student.getName();
        student.getAge();
        student.printThrowException();
    }
}

```



## 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"
    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 ">

    <aop:config>
        <aop:aspect id = "log" ref = "logging">
            <aop:pointcut id = "selectAll"
                expression = "execution(* com.tutorialspoint.*.*(..))"/>

            <aop:before pointcut-ref = "selectAll" method = "beforeAdvice"/>
            <aop:after pointcut-ref = "selectAll" method = "afterAdvice"/>
            <aop:after-returning pointcut-ref = "selectAll"
                returning = "retVal" method = "afterReturningAdvice"/>

            <aop:after-throwing pointcut-ref = "selectAll"
                throwing = "ex" method = "AfterThrowingAdvice"/>
        </aop:aspect>
    </aop:config>

    <!-- Definition for student bean -->
    <bean id = "student" class = "com.tutorialspoint.Student">
        <property name = "name" value = "Zara" />
        <property name = "age" value = "11"/>
    </bean>

    <!-- Definition for logging aspect -->
    <bean id = "logging" class = "com.tutorialspoint.Logging"/>

</beans>
```

## Output

```
Going to setup student profile.
Name : Zara
Student profile has been setup.
Returning:Zara
Going to setup student profile.
Age : 11
Student profile has been setup.
Returning:11
Going to setup student profile.
Exception raised
Student profile has been setup.
There has been an exception: java.lang.IllegalArgumentException
.....other exception content
```

## Practical 8.1

---

Spring AOP – before advice

---

Write a program to demonstrate Spring AOP – before advice.

### Logging.java

```
package com.hiraymca;
public class Logging {

    //Types of Advice
    //1 beforeAdvice
    public void beforeAdvice()
    {
        System.out.println("Setuping Student Profile ");
    }
}
```

### 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) {
        // TODO Auto-generated method 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:before method="beforeAdvice" 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

---

Setuping Student Profile  
Name=Sukhiram  
Setuping Student Profile  
Age-35

## Practical 8.2

---

Spring AOP – after advice

---

Write a program to demonstrate Spring AOP – after advice.

### Logging.java

```
package com.hiraymca;
public class Logging {

    //Types of Advice
    //1 beforeAdvice
    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) {
        // TODO Auto-generated method 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 9.1

---

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

### **Student Interface      iStudent.java**

---

```
package com.hiraymca;
import java.util.List;
import javax.sql.DataSource;
public interface IStudent {
    //setting data source
    public void setDataSource(DataSource datasource);

    //creating records- inserting record into table

    public void create(int rollno,String name, int age);

    //reading specific records
    public Student readStudent(int id);

    //reading all the records from table
    public List<Student> listStudents();

    //update record
    public void update(int id,int rollno,String name, int age);

    //delete record
    public void delete(int id);
}
```



## Student.java

---

```
package com.hiraymca;

import java.util.List;
import javax.sql.DataSource;

public class Student {
    int id;
    int rollno;
    String sname;
    int age;
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public int getRollno() {
        return rollno;
    }
    public void setRollno(int rollno) {
        this.rollno = rollno;
    }
    public String getSname() {
        return sname;
    }
    public void setSname(String sname) {
        this.sname = sname;
    }
}
```

```

    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
}

```

### **StudentJDBCTemplate.java**

---

```

package com.hiraymca;
import java.util.List;

import javax.sql.DataSource;
import org.springframework.jdbc.core.JdbcTemplate;

public class StudentJDBCTemplate implements IStudent {
    private DataSource dataSource;
    private JdbcTemplate jdbcTemplateObject;

    @Override
    public void setDataSource(DataSource datasource) {
        // TODO Auto-generated method stub
        this.dataSource=datasource;
        this.jdbcTemplateObject=new JdbcTemplate(datasource);
    }

    @Override

```

```

//inserting a new record into database
public void create(int rollno, String name, int age) {
    // TODO Auto-generated method stub
    String sql="insert into student2(rollno,sname,age)values(?,?,?)";
    jdbcTemplateObject.update(sql,rollno,name,age);
    System.out.println("Record inserted successfully");
}

@Override
public Student readStudent(int id) {
    // TODO Auto-generated method stub
    String sql="select * from student2 where id=?";
    Student
student=(Student)jdbcTemplateObject.queryForObject(sql,new Object[]
{id},new StudentMapper());
    return student;
}

@Override
public List<Student> listStudents() {
    // TODO Auto-generated method stub
    String sql="select * from student2";
    List<Student> students=jdbcTemplateObject.query(sql, new
StudentMapper());
    return students;
}

@Override

```

```

        public void update(int id, int rollno, String sname, int age) {
            // TODO Auto-generated method stub
            String SQL = "update Student2 set rollno=?,sname=?,age = ?
where id = ?";
            jdbcTemplateObject.update(SQL, rollno,sname,age,id);
            System.out.println("Updated Record with ID = " + id );
            return;
        }

        @Override
        public void delete(int id) {
            // TODO Auto-generated method stub
            String SQL = "delete from Student2 where id = ?";
            jdbcTemplateObject.update(SQL, id);
            System.out.println("Deleted Record with ID = " + id );
        }
    }

```

### **StudentMapper.java**

---

```

package com.hiraymca;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;
import javax.sql.DataSource;
import org.springframework.jdbc.core.RowMapper;
public class StudentMapper implements RowMapper<Student>
{
    public Student mapRow(ResultSet rs, int rowNum) throws
SQLException {

```

```

        Student student = new Student();
        student.setId(rs.getInt("id"));
        student.setRollno(rs.getInt("rollno"));
        student.setSname(rs.getString("sname"));
        student.setAge(rs.getInt("age"));
        return student;
    }
}

```

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

    <bean id="dataSource"
        class = "org.springframework.jdbc.datasource.DriverManagerDataSource">
        <property name = "driverClassName" value = "com.mysql.jdbc.Driver"/>
        <property name = "url" value =
"jdbc:mysql://localhost:3306/studentdb"/>
        <property name = "username" value = "root"/>
        <property name = "password" value = ""/>
    </bean>

    <bean id="studentJdbcTemplate" class="com.hiraymca.StudentJdbcTemplate">
        <property name='dataSource' ref="dataSource"/></property>
    </bean>

</beans>

```

## MainApp.java

---

```
package com.hiraymca;
import java.util.List;
import java.util.Scanner;
import java.util.ArrayList;

import org.springframework.context.ApplicationContext;
import
org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.dao.EmptyResultDataAccessException;

public class MainApp {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        ApplicationContext context=new
ClassPathXmlApplicationContext("Beans.xml");
        StudentJDBCTemplate studentJDBCTemplate;
        studentJDBCTemplate=(StudentJDBCTemplate)context.getBean("stude
ntJDBCTemplate");
        int choice;
        int id;
        int rollno;
        String sname;
        int age;
        String ans=null;
        Scanner sc=new Scanner(System.in);
        do
```

```

{
    System.out.println("Select the choice");
    System.out.println("1.Insert record ");
    System.out.println("2.Read record");
    System.out.println("3.List all the records");
    System.out.println("4.Delete the record");
    System.out.println("5.Exit");
    System.out.print("Enter your choice(1..5):");
    choice=sc.nextInt();
    switch(choice)
    {
        case 1:
            //inserting a record
            System.out.print("Enter roll no");
            rollno=sc.nextInt();
            System.out.println("Enter name");
            sname=sc.next();
            System.out.println("Enter age");
            age=sc.nextInt();
            studentJDBCTemplate.create(rollno, sname, age);
            break;
        case 2:
            //reading a record
            Student student;
            System.out.println("Enter record id");
            id=sc.nextInt();
            try
            {

```

```

        student=studentJDBCTemplate.readStudent(id);
        System.out.println("id="+student.getId());
        System.out.println("Rollno="+student.rollno);
        System.out.println("Name="+student.getSname());
        System.out.println("Age="+student.getAge());
    }catch(Exception ex)
    {
        System.out.println("Record not found");
    }
        break;
    case 3:
        //listing all the records
        List<Student>
students=studentJDBCTemplate.listStudents();

        System.out.println("id"+"\\t"+"Rollno"+"\\t"+"Name"+"\\t\\t"+"Age");
        for (Student record : students) {

            System.out.println(record.getId()+"\\t"+record.getRollno()+"\\t"+record.
getSname()+"\\t\\t"+record.getAge());
        }
        break;
    case 4:
        // delete the record
        System.out.println("Enter record id");
        id=sc.nextInt();
        try
        {

```



```

        studentJDBCTemplate.delete(id);
    }
    catch(Exception ex)
    {
        System.out.println("Record not found");
    }
    break;
case 5:
    System.exit(0);
    break;
}
System.out.println("Do you wish to continue(y/n)");
ans=sc.next();
}while(ans.equals("Y") || ans.equals("y"));

}

}

```

## Practical 9.2

---

Write a program to demonstrate PreparedStatement in Spring JdbcTemplate

### IStudent.java

---

```
package com.hiraymca;
java.util.List;
import javax.sql.DataSource;
public interface IStudent {
    //setting data source
    public void setDataSource(DataSource datasource);

    //creating records- inserting record into table
    public void create(int rollno,String name, int age);

    //listing all records
    public List<Student> listStudents();
}
```

### Student.java

---

```
package com.hiraymca;
import java.util.List;
import javax.sql.DataSource;

public class Student {
    int id;
    int rollno;
    String sname;
    int age;

    public Student(int rollno, String sname, int age) {

        this.rollno = rollno;
        this.sname = sname;
        this.age = age;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
```

```

        this.id = id;
    }

    public int getRollno() {
        return rollno;
    }

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

    public String getSname() {
        return sname;
    }

    public void setSname(String sname) {
        this.sname = sname;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }
}

```

#### StudentDAO.java

---

```

package com.hiraymca;
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 StudentDAO {
    private JdbcTemplate jdbcTemplate;

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

    public Boolean saveStudent(final Student s)

```

```

        {
            String query="insert into
student2(rollno,sname,age)values(?,?,?)" ;
            return jdbcTemplate.execute(query,new
PreparedStatementCallback<Boolean>() {
                @Override
                public Boolean doInPreparedStatement(PreparedStatement
ps)

                    throws SQLException, DataAccessException {

                    ps.setInt(1,s.getRollno());
                    ps.setString(2,s.getSname());
                    ps.setFloat(3,s.getAge());

                    return ps.execute();

                }

            });
        }
    }
}

```

#### MainApp.java

---

```

package com.hiraymca;
import java.util.List;
import java.util.Scanner;
import java.util.ArrayList;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.dao.EmptyResultDataAccessException;

public class MainApp {
    private static int rollno;
    private static String sname;
    private static int age;
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        ApplicationContext context=new ClassPathXmlApplicationContext("Beans.xml");
        Scanner sc=new Scanner(System.in);
        StudentDAO stud=(StudentDAO)context.getBean("studentDAO");
        System.out.println("Enter rollno");
    }
}

```

```

        rollno=sc.nextInt();
        System.out.println("Enter student name");
        sname=sc.next();
        System.out.println("Enter age");
        age=sc.nextInt();
        stud.saveStudent(new Student(rollno,sname,age));
        System.out.println("Record inserted successfully");
    }
}

```

#### 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"
       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 name = "url" value =
"jdbc:mysql://localhost:3306/studentdb"/>
        <property name = "username" value = "root"/>
        <property name = "password" value = ""/>
    </bean>
    <bean id="jdbcTemplate"
class="org.springframework.jdbc.core.JdbcTemplate">
        <property name="dataSource" ref="ds"></property>
    </bean>
    <bean id="studentDAO" class="com.hiraymca.StudentDAO">
        <property name='jdbcTemplate' ref="jdbcTemplate"></property>
    </bean>
</beans>

```

#### Output

---

```

Enter rollno 4
Enter student name ddd
Enter age 25
Record inserted successfully

```

## Practical 9.3

---

Write a program in Spring JDBC to demonstrate ResultSetExtractor Interface

iStudent.java [Student interface]

---

```
package com.hiraymca;

import java.util.List;

import javax.sql.DataSource;

public interface IStudent {

    //setting data source

    public void setDataSource(DataSource datasource);

    //creating records- inserting record into table

    public void create(int rollno,String name, int age);

    //listing all records

    public List<Student> listStudents();

}
```

Student.java

---

```
package com.hiraymca;

import java.util.List;

import javax.sql.DataSource;

public class Student {

    int id;
    int rollno;
    String sname;
    int age;
    public Student()
    {
    }
    public Student(int rollno, String sname, int age) {

        this.rollno = rollno;
        this.sname = sname;
        this.age = age;
    }

}
```

```

        public int getId() {
            return id;
        }

        public void setId(int id) {
            this.id = id;
        }

        public int getRollno() {
            return rollno;
        }

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

        public String getSname() {
            return sname;
        }

        public void setSname(String sname) {
            this.sname = sname;
        }

        public int getAge() {
            return age;
        }

        public void setAge(int age) {
            this.age = age;
        }
    }

```

StudentDAO.java

---

```

package com.hiraymca;

import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import java.sql.ResultSet;
import org.springframework.dao.DataAccessException;

```

```

import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.jdbc.core.PreparedStatementCallback;
import org.springframework.jdbc.core.ResultSetExtractor;

public class StudentDAO {
    private JdbcTemplate jdbcTemplate;

    public void setJdbcTemplate(JdbcTemplate jdbcTemplate) {
        this.jdbcTemplate = jdbcTemplate;
    }
    public Boolean saveStudent(final Student s)
    {
        String query="insert into
student2(rollno,sname,age)values(?,?,?)"
        return jdbcTemplate.execute(query,new
PreparedStatementCallback<Boolean>() {
            @Override
            public Boolean doInPreparedStatement(PreparedStatement
ps)

                throws SQLException, DataAccessException {

                ps.setInt(1,s.getRollno());
                ps.setString(2,s.getSname());
                ps.setFloat(3,s.getAge());

                return ps.execute();
            }

        });
    }
    public List<Student> getAllStudents(){
        return jdbcTemplate.query("select * from student2",
new ResultSetExtractor<List<Student>>() {

            @Override
            public List<Student> extractData(ResultSet rs)
throws SQLException, DataAccessException {
                // TODO Auto-generated method stub
                List<Student>list=new ArrayList<Student>();
                while(rs.next())
                {
                    Student s=new Student();
                    s.setId(rs.getInt(1));

```



```

        s.setRollno(rs.getInt(2));
        s.setSname(rs.getString(3));
        s.setAge(rs.getInt(4));
        list.add(s);
    }
    return list;
}

});
}
}
MainApp.java
package com.hiraymca;
import java.util.List;
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Iterator;

import org.springframework.context.ApplicationContext;
import
org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.dao.EmptyResultDataAccessException;

public class MainApp {
    private static int rollno;
    private static String sname;
    private static int age;
    private static int choice;
    private static String ans;
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        ApplicationContext context=new
ClassPathXmlApplicationContext("Beans.xml");
        Scanner sc=new Scanner(System.in);
        StudentDAO stud=(StudentDAO)context.getBean("studentDAO");
        do
        {
            System.out.println("1.Insert record");
            System.out.println("2.List all record");
            System.out.println("3.Exit");
            System.out.print("Enter your choice(1..3)");
            choice=sc.nextInt();
            switch(choice)
            {
                case 1:

```

```

        //inserting record
        System.out.print("Enter rollno");
        rollno=sc.nextInt();
        System.out.print("Enter student name");
        sname=sc.next();
        System.out.print("Enter age");
        age=sc.nextInt();
        stud.saveStudent(new Student(rollno,sname,age));
        System.out.println("Record inserted successfully");
        break;
    case 2:
        //listing record
        List<Student>slist=stud.getAllStudents();
        System.out.println("Rollno"+"\\t"+"Name"+"\\t"+"Age");
        for (Student student : slist) {

            System.out.println(student.getRollno()+"\\t"+student.getSname()+"\\t"+student.getAge());

        }
        break;
    case 3:
        //exiting from application
        System.exit(0);
    }
    System.out.print("Do you wish to continue(y/n) ");
    ans=sc.next();
}while(ans.equals("y")||ans.equals("Y"));
}

}
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"
       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 name = "url" value =
"jdbc:mysql://localhost:3306/studentdb"/>

```

```

        <property name = "username" value = "root"/>
        <property name = "password" value = ""/>
    </bean>
    <bean id="jdbcTemplate"
class="org.springframework.jdbc.core.JdbcTemplate">
        <property name="dataSource" ref="ds"></property>
    </bean>
    <bean id="studentDAO" class="com.hiraymca.StudentDAO">
        <property name='jdbcTemplate' ref="jdbcTemplate"></property>
    </bean>

</beans>

```

## Output

---

```

1.Insert record
2.List all record
3.Exit
Enter your choice(1..3)2
Rollno      Name  Age
1          aaa   23
2          bbb   21
3          ccc   23
4          ddd   25
5          eee   21
6          fff   23
7          ggg   21
Do you wish to cotinue(y/n)

```

## Practical 9.4

---

Write a program in Spring JDBC to demonstrate ResultSetExtractor Interface

iStudent.java [Student interface]

---

```
package com.hiraymca;

import java.util.List;

import javax.sql.DataSource;

public interface IStudent {

    //setting data source

    public void setDataSource(DataSource datasource);

    //creating records- inserting record into table

    public void create(int rollno,String name, int age);

    //listing all records

    public List<Student> listStudents();

}
```

Student.java

---

```
package com.hiraymca;

import java.util.List;

import javax.sql.DataSource;

public class Student {

    int id;
    int rollno;
    String sname;
    int age;
    public Student()
    {
    }
    public Student(int rollno, String sname, int age) {

        this.rollno = rollno;
        this.sname = sname;
        this.age = age;
    }

}
```

```

        public int getId() {
            return id;
        }

        public void setId(int id) {
            this.id = id;
        }

        public int getRollno() {
            return rollno;
        }

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

        public String getSname() {
            return sname;
        }

        public void setSname(String sname) {
            this.sname = sname;
        }

        public int getAge() {
            return age;
        }

        public void setAge(int age) {
            this.age = age;
        }
    }
}

StudentDAO.java
package com.hiraymca;

import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import java.sql.ResultSet;
import org.springframework.dao.DataAccessException;

```

---

```

import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.jdbc.core.PreparedStatementCallback;
import org.springframework.jdbc.core.ResultSetExtractor;

public class StudentDAO {
    private JdbcTemplate jdbcTemplate;

    public void setJdbcTemplate(JdbcTemplate jdbcTemplate) {
        this.jdbcTemplate = jdbcTemplate;
    }
    public Boolean saveStudent(final Student s)
    {
        String query="insert into
student2(rollno,sname,age)values(?,?,?)"
        return jdbcTemplate.execute(query,new
PreparedStatementCallback<Boolean>() {
            @Override
            public Boolean doInPreparedStatement(PreparedStatement
ps)
                throws SQLException, DataAccessException {

                ps.setInt(1,s.getRollno());
                ps.setString(2,s.getSname());
                ps.setFloat(3,s.getAge());

                return ps.execute();

            }

        });
    }

    public List<Student> listStudents() {
        // TODO Auto-generated method stub
        String sql="select * from student2";
        List<Student> students=jdbcTemplate.query(sql, new
StudentMapper());
        return students;
    }
}

```

---

MainApp.java

---

```
package com.hiraymca;
import java.util.List;
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Iterator;

import org.springframework.context.ApplicationContext;
import
org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.dao.EmptyResultDataAccessException;

public class MainApp {
    private static int rollno;
    private static String sname;
    private static int age;
    private static int choice;
    private static String ans;
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        ApplicationContext context=new
ClassPathXmlApplicationContext("Beans.xml");
        Scanner sc=new Scanner(System.in);
        StudentDAO stud=(StudentDAO)context.getBean("studentDAO");
        do
        {
            System.out.println("1.Insert record");
            System.out.println("2.List all record");
            System.out.println("3.Exit");
            System.out.print("Enter your choice(1..3)");
            choice=sc.nextInt();
            switch(choice)
            {
                case 1:
                    //inserting record
                    System.out.print("Enter rollno");
                    rollno=sc.nextInt();
                    System.out.print("Enter student name");
                    sname=sc.next();
                    System.out.print("Enter age");
                    age=sc.nextInt();
                    stud.saveStudent(new Student(rollno,sname,age));
                    System.out.println("Record inserted successfully");
                    break;
                case 2:
```

```

        //listing record
        List<Student>slst=stud.listStudents();
        System.out.println("Rollno"+"\\t"+"Name"+"\\t"+"Age");
        for (Student student : slst) {

            System.out.println(student.getRollno()+"\\t"+student.getSname()+"\\t"+student.getAge());

        }
        break;
    case 3:
        //exiting from application
        System.exit(0);
    }
    System.out.print("Do you wish to continue(y/n) ");
    ans=sc.next();
}while(ans.equals("y")||ans.equals("Y"));
}

```

}  
 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"
        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 name = "url" value =
        "jdbc:mysql://localhost:3306/studentdb"/>
        <property name = "username" value = "root"/>
        <property name = "password" value = ""/>
    </bean>
    <bean id="jdbcTemplate"
    class="org.springframework.jdbc.core.JdbcTemplate">
        <property name="dataSource" ref="ds"></property>
    </bean>
    <bean id="studentDAO" class="com.hiraymca.StudentDAO">
        <property name='jdbcTemplate' ref="jdbcTemplate"></property>
    </bean>

```



</beans>

## Output

---

```
1.Insert record
2.List all record
3.Exit
Enter your choice(1..3)2
Rollno      Name  Age
1          aaa   23
2          bbb   21
3          ccc   23
4          ddd   25
5          eee   21
6          fff   23
7          ggg   21
Do you wish to continue(y/n)
```

## Practical 10.1

---

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;

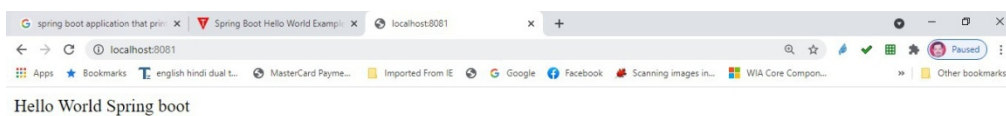
@RestController
@EnableAutoConfiguration
@SpringBootApplication
public class SpringBootDemoApplication {

    public static void main(String[] args) {
        SpringApplication.run(SpringBootDemoApplication.class,
args);
    }
    @RequestMapping("/")
    public String home()
    {
        return "Hello World Spring boot";
    }

}
```

### Output

---



## Practical 10.2

---

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
        successssfully", 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
        successssfully", 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
        successssfully", 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);

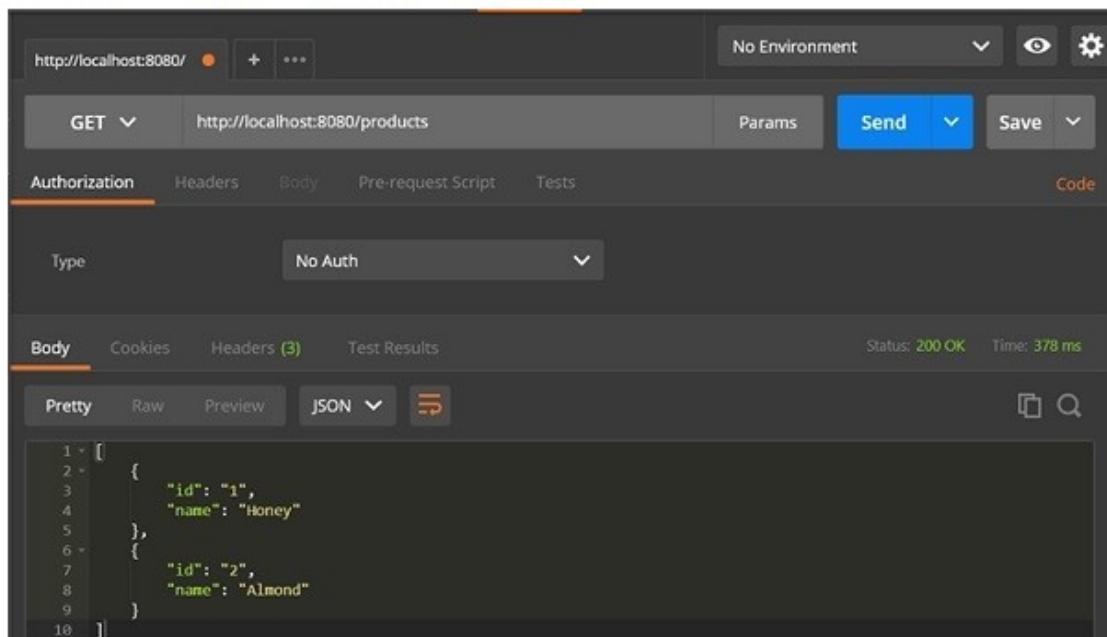
    }

}
```

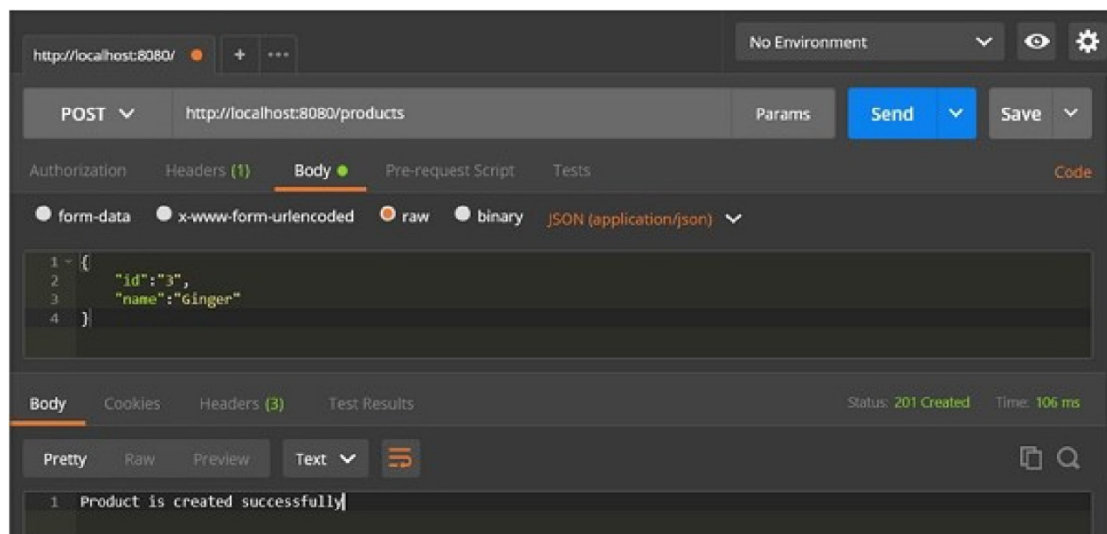
## Output

---

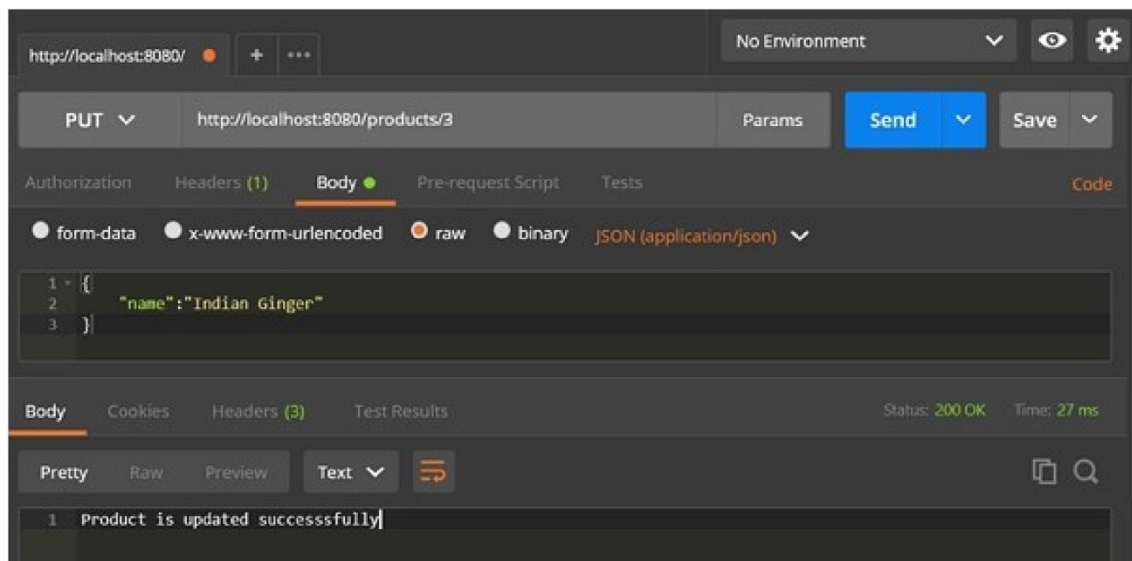
GET API URL is: **http://localhost:8080/products**



POST API URL is: **http://localhost:8080/products**



PUT API URL is: **http://localhost:8080/products/3**



DELETE API URL is: **http://localhost:8080/products/3**

